

# Automates programmables Twido

## Guide de mise en œuvre logicielle

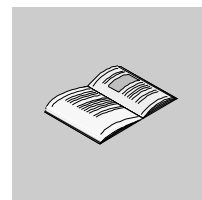
TWD USE 10AF fre Version 2.5





---

# Table des matières



	<b>Consignes de sécurité</b>	<b>11</b>
	<b>A propos de ce manuel</b>	<b>15</b>
<b>Partie I</b>	<b>Description du logiciel Twido</b>	<b>17</b>
	Présentation	17
<b>Chapitre 1</b>	<b>Introduction au logiciel Twido</b>	<b>19</b>
	Présentation	19
	Introduction à TwidoSoft	20
	Introduction aux langages Twido	21
<b>Chapitre 2</b>	<b>Objets langage Twido</b>	<b>25</b>
	Présentation	25
	Validation d'un objet langage	26
	Objets bits	27
	Objets mots	30
	Objets flottants et mots doubles	34
	Adressage d'objets bits	39
	Adressage d'objets mots	40
	Adressage d'objets flottants	41
	Adressage d'objets mots doubles	42
	Repérage des entrées/sorties	43
	Adressage réseau	45
	Objets blocs fonction	46
	Objets structurés	48
	Objets indexés	52
	Symbolisation d'objets	54
<b>Chapitre 3</b>	<b>Mémoire utilisateur</b>	<b>55</b>
	Présentation	55
	Structure de la mémoire utilisateur	56
	Sauvegarde et restauration sans cartouche de sauvegarde, ni cartouche de mémoire étendue	59
	Sauvegarde et restauration avec une cartouche de sauvegarde de 32 K	61

---

	Utilisation de la cartouche de mémoire étendue de 64 K .....	64
<b>Chapitre 4</b>	<b>Modes de fonctionnement de l'automate .....</b>	<b>67</b>
	Présentation .....	67
	Scrutation cyclique .....	68
	Scrutation périodique .....	70
	Vérification de la durée de scrutation .....	73
	Modes de fonctionnement .....	75
	Gestion des coupures et des reprises secteur .....	77
	Gestion d'une reprise à chaud .....	80
	Gestion d'un démarrage à froid .....	82
	Initialisation des objets .....	84
<b>Chapitre 5</b>	<b>Gestion des tâches événementielles .....</b>	<b>85</b>
	En bref... ..	85
	Présentation des tâches événementielles .....	86
	Description des différentes sources d'événement .....	87
	Gestion des événements .....	89
<b>Partie II</b>	<b>Fonctions spéciales .....</b>	<b>91</b>
	Présentation .....	91
<b>Chapitre 6</b>	<b>Communications .....</b>	<b>93</b>
	Présentation .....	93
	Présentation des différents types de communications .....	95
	Communications entre TwidoSoft et l'automate .....	97
	Communication entre TwidoSoft et un modem .....	103
	Communications de liaison distante .....	114
	Communications ASCII .....	127
	Communications Modbus .....	138
	Requêtes Modbus standard .....	153
	Vue d'ensemble des communications TCP/IP Ethernet .....	159
	Guide de configuration rapide TCP/IP pour les communications Ethernet PC vers automate .....	160
	Connexion de l'automate au réseau .....	166
	Adressage IP .....	167
	Affectation d'adresses IP .....	169
	Configuration TCP/IP .....	173
	Onglet Configurer adresse IP .....	175
	Onglet IP repérée .....	177
	Onglet Contrôle d'inactivité .....	179
	Onglet Périphériques distants .....	181
	Affichage de la configuration Ethernet .....	183
	Gestion des connexions Ethernet .....	185
	Voyants Ethernet .....	188
	Messagerie Modbus TCP .....	190

---

<b>Chapitre 7</b>	<b>Fonctions analogiques intégrées</b>	<b>195</b>
	Présentation	195
	Point de réglage analogique	196
	Voie analogique.	198
<b>Chapitre 8</b>	<b>Gestion des modules analogiques</b>	<b>199</b>
	Présentation	199
	Présentation des modules analogiques	200
	Adressage d'entrées et de sorties analogiques	201
	Configuration d'entrées et de sorties analogiques	203
	Informations sur l'état du module analogique	205
	Exemples d'utilisation de modules analogiques	206
<b>Chapitre 9</b>	<b>Mise en œuvre du bus AS-Interface V2</b>	<b>209</b>
	Présentation	209
	Présentation du bus AS-Interface V2	210
	Description fonctionnelle générale	211
	Principes de mise en œuvre logicielle.	214
	Description de l'écran de configuration du bus AS-Interface	216
	Configuration du bus AS-Interface	218
	Description de l'écran de mise au point	224
	Modification de l'adresse d'un esclave	227
	Mise à jour de la configuration du bus AS-Interface en mode connecté.	229
	Adressage automatique d'un esclave AS-Interface V2.	234
	Comment insérer un équipement esclave dans une configuration AS-Interface V2 existante	235
	Remplacement automatique d'un esclave AS-Interface V2 défectueux	236
	Adressage des entrées/sorties associées aux équipements esclaves connectés sur bus AS-Interface V2	237
	Programmation et diagnostic du bus AS-Interface V2	239
	Mode de fonctionnement du module interface bus AS-Interface V2.	244
<b>Chapitre 10</b>	<b>Fonctionnement de l'afficheur.</b>	<b>245</b>
	Présentation	245
	Afficheur	246
	Informations d'identification et états de l'automate	249
	Variables et objets système	251
	Paramètres de port série.	258
	Horloge calendrier	259
	Facteur de correction de l'horodateur	260
<b>Partie III</b>	<b>Description des langages Twido</b>	<b>261</b>
	Présentation	261
<b>Chapitre 11</b>	<b>Langage schéma à contacts</b>	<b>263</b>
	Présentation	263

---

	Introduction aux schémas à contacts . . . . .	264
	Principes de programmation en langage schéma à contacts . . . . .	266
	Blocs de schémas à contacts . . . . .	268
	Éléments graphiques du langage schéma à contacts . . . . .	271
	Instructions spéciales OPEN et SHORT du langage schéma à contacts . . . . .	274
	Conseils de programmation . . . . .	275
	Réversibilité schéma à contacts/liste . . . . .	280
	Recommandations pour la réversibilité entre le langage schéma à contacts et le langage liste d'instructions . . . . .	282
	Documentation du programme . . . . .	284
<b>Chapitre 12</b>	<b>Langage liste d'instructions . . . . .</b>	<b>287</b>
	Présentation . . . . .	287
	Vue d'ensemble des programmes en langage liste d'instructions . . . . .	288
	Fonctionnement des listes d'instructions . . . . .	290
	Instructions en langage liste d'instructions . . . . .	291
	Utilisation de parenthèses . . . . .	294
	Instructions de pile (MPS, MRD, MPP) . . . . .	297
<b>Chapitre 13</b>	<b>Grafcet . . . . .</b>	<b>299</b>
	Présentation . . . . .	299
	Description des instructions Grafcet . . . . .	300
	Description de la structure d'un programme Grafcet . . . . .	305
	Actions associées aux étapes Grafcet . . . . .	308
<b>Partie IV</b>	<b>Description des instructions et des fonctions . . . . .</b>	<b>311</b>
	Présentation . . . . .	311
<b>Chapitre 14</b>	<b>Instructions élémentaires . . . . .</b>	<b>313</b>
	Présentation . . . . .	313
14.1	Traitement booléen . . . . .	315
	Présentation . . . . .	315
	Instructions booléennes . . . . .	316
	Explication du format de description des instructions booléennes . . . . .	318
	Instructions de chargement (LD, LDN, LDR, LDF) . . . . .	320
	Instructions d'affectation (ST, STN, R, S) . . . . .	322
	Instructions AND logique (AND, ANDN, ANDR, ANDF) . . . . .	324
	Instructions OR logique (OR, ORN, ORR, ORF) . . . . .	326
	OR exclusif, instructions (XOR, XORN, XORR, XORF) . . . . .	328
	Instruction NOT (N) . . . . .	330
14.2	Blocs fonctions élémentaires . . . . .	332
	Présentation . . . . .	332
	Blocs fonctions standards . . . . .	333
	Principes de programmation de blocs fonction standards . . . . .	335
	Bloc fonction temporisateur (%Tmi) . . . . .	337
	Type de temporisateur TOF . . . . .	339

	Type de temporisateur TON .....	340
	Type de temporisateur TP .....	341
	Programmation et configuration de temporisateurs .....	342
	Bloc fonction compteur/décompteur (%Ci) .....	345
	Programmation et configuration des compteurs .....	349
	Bloc fonction registre bits à décalage (%SBRi) .....	351
	Bloc fonction pas à pas (%SCi) .....	354
14.3	Traitement numérique .....	358
	Présentation .....	358
	Introduction aux instructions numériques .....	359
	Instructions d'affectation .....	360
	Instructions de comparaison .....	365
	Instructions arithmétiques sur entiers .....	367
	Instructions logiques .....	370
	Instructions de décalage .....	372
	Instructions de conversion .....	374
	Instructions de conversion entre mots simples et doubles .....	376
14.4	Instructions sur programme .....	377
	Présentation .....	377
	Instructions END .....	378
	Instruction NOP .....	380
	Instructions de saut .....	381
	Instructions de sous-programme .....	382
<b>Chapitre 15</b>	<b>Instructions avancées .....</b>	<b>385</b>
	Présentation .....	385
15.1	Blocs fonctions avancées .....	387
	Présentation .....	387
	Objets mots et objets bits associés à des blocs fonction avancés .....	388
	Principes de programmation de blocs fonctions avancés .....	390
	Bloc fonction registre LIFO/FIFO (%Ri) .....	393
	LIFO, fonctionnement .....	395
	FIFO, fonctionnement .....	396
	Programmation et configuration des registres .....	397
	Bloc fonction %PWM (modulation de la largeur d'impulsion) .....	400
	Bloc fonction sortie du générateur d'impulsions (%PLS) .....	404
	Bloc fonction programmeur cyclique (%DR) .....	408
	Fonctionnement du bloc fonction programmeur cyclique %DRi .....	410
	Programmation et configuration des programmeurs cycliques .....	412
	Bloc fonction compteur rapide (%FC) .....	414
	Bloc fonction compteur rapide (%VFC) .....	418
	Emission/réception de messages - Instruction d'échange (EXCH) .....	431
	Bloc fonction de contrôle d'échange (%MSGx) .....	432
15.2	Fonctions horodateur .....	436
	Présentation .....	436
	Fonctions horloges .....	437

---

	Blocs horodateurs . . . . .	438
	Horodatage . . . . .	441
	Réglage de la date et de l'heure . . . . .	443
15.3	Fonction PID . . . . .	448
	Présentation . . . . .	448
	Présentation générale . . . . .	450
	Principe de la boucle de régulation . . . . .	451
	Méthodologie de développement d'une application de régulation . . . . .	452
	Compatibilités et performances . . . . .	453
	Caractéristiques détaillées de la fonction PID . . . . .	454
	Comment accéder à la configuration du PID . . . . .	458
	Onglet Général du PID . . . . .	460
	Onglet Entrée du PID . . . . .	463
	Onglet PID . . . . .	466
	Onglet Auto tuning de la fonction PID . . . . .	469
	Onglet Sortie du PID . . . . .	474
	Comment accéder à la mise au point du PID . . . . .	477
	Onglet Animation du PID . . . . .	479
	Onglet Trace du PID . . . . .	482
	Etats du PID et codes d'erreurs . . . . .	485
	Réglage PID avec la fonction d'auto tuning (AT) . . . . .	489
	Méthode de réglage des paramètres PID . . . . .	500
	Rôle et influence des paramètres d'un PID . . . . .	504
	Annexe 1 : Notions fondamentales de la théorie PID . . . . .	508
	Annexe 2 : Premier ordre avec modèle de temporisation . . . . .	510
15.4	Instructions sur flottants . . . . .	512
	Présentation . . . . .	512
	Instructions arithmétiques sur flottant . . . . .	513
	Instructions Trigonométriques . . . . .	518
	Instructions de conversion . . . . .	521
	Instructions de conversion Entier <-> Flottant . . . . .	523
15.5	Instructions sur tableaux d'objets . . . . .	526
	Présentation . . . . .	526
	Fonction de sommation sur tableaux . . . . .	527
	Fonction de comparaison de tableaux . . . . .	529
	Fonctions de recherche sur tableaux . . . . .	531
	Fonctions de recherche de valeurs maxi et mini sur tableaux . . . . .	533
	Nombre d'occurrences d'une valeur dans un tableau . . . . .	534
	Fonction décalage circulaire sur un tableau . . . . .	535
	Fonction de tri sur tableau . . . . .	537
	Fonction d'interpolation sur tableau de flottants . . . . .	538
	Fonction de moyenne des valeurs d'un tableau de flottants . . . . .	543

---



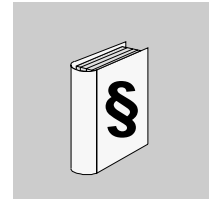
---

<b>Chapitre 16</b>	<b>Bits système et mots système</b>	<b>545</b>
	Présentation	545
	Bits système (%S)	546
	Mots système (%SW)	555
<b>Glossaire</b>		<b>569</b>
<b>Index</b>		<b>583</b>



---

## Consignes de sécurité



---

### Informations importantes

#### AVIS

Lisez attentivement ces instructions et familiarisez-vous avec le matériel avant d'essayer de l'installer, de le faire fonctionner ou d'effectuer une opération de maintenance. Les messages spéciaux qui suivent peuvent apparaître partout dans ce document ou sur l'appareil. Ils vous avertissent de dangers potentiels ou attirent votre attention sur des renseignements pouvant éclairer ou simplifier une procédure.



La présence de ce symbole sur une étiquette de danger ou d'avertissement indique qu'un risque d'électrocution existe, pouvant provoquer des lésions corporelles si les instructions ne sont pas respectées.



Ceci est le symbole d'une alerte de sécurité. Il sert à vous avertir d'un danger potentiel de blessures corporelles. Respectez toutes les consignes de sécurité accompagnant ce symbole pour éviter toute situation potentielle de blessure ou de mort.



### DANGER

La mention DANGER signifie qu'il existe une situation potentiellement dangereuse qui, n'est pas évitée, **entraînera** la mort, des blessures graves ou des dommages matériels.



### AVERTISSEMENT

La mention AVERTISSEMENT signifie qu'il existe une situation potentiellement dangereuse qui, si elle n'est pas évitée, **peut entraîner** la mort, des blessures graves ou des dommages matériels.



### ATTENTION

La mention ATTENTION signifie qu'il existe une situation potentiellement dangereuse qui, si elle n'est pas évitée, **peut entraîner** des lésions corporelles ou des dommages matériels.

**VEUILLEZ  
REMARQUER**

L'entretien du matériel électrique ne doit être effectué que par du personnel qualifié. Schneider Electric n'assume aucune responsabilité concernant les conséquences éventuelles découlant de l'utilisation de cette documentation. Ce document n'est pas destiné à servir de manuel d'utilisation aux personnes sans formation. Le manuel de référence du matériel Twido, TWD USE 10AF, contient les instructions d'assemblage et d'installation.  
(c) 2002-2004 Schneider Electric Tous droits réservés


---


**Informations  
supplémentaires  
relatives à la  
sécurité**

Les personnes chargées de l'application, de la mise en œuvre ou de l'utilisation de ce produit doivent s'assurer que les principes de conception nécessaires ont été inclus dans chacune des applications, en totale conformité avec les normes, codes, règlements, exigences en matière de performance et de sécurité et lois en vigueur.

---

**Avertissements  
généraux et  
précautions à  
prendre**

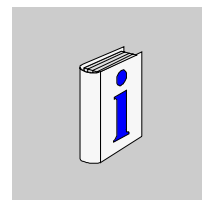
	<b>AVERTISSEMENT</b>
	<b>RISQUE D'EXPLOSION</b> <ul style="list-style-type: none"> <li>• Le remplacement de composants risque d'affecter la conformité de l'équipement à la Classe 1, Division 2.</li> <li>• Assurez-vous que l'alimentation est coupée ou que la zone ne présente aucun danger avant de déconnecter l'équipement.</li> </ul> <p><b>Le non-respect de ces précautions peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.</b></p>

	<b>AVERTISSEMENT</b>
	<b>FONCTIONNEMENT ACCIDENTEL DE L'EQUIPEMENT</b> <ul style="list-style-type: none"> <li>• Coupez l'alimentation avant de procéder à tout retrait, installation, câblage, entretien et contrôle.</li> <li>• Ce produit n'est pas conçu pour être utilisé lors d'opérations dangereuses pour la sécurité. Lorsque des risques de lésions corporelles ou de dommages matériels existent, utilisez les verrous de sécurité câblés appropriés.</li> <li>• Ne pas désassembler, réparer ou modifier les modules.</li> <li>• Cet automate est conçu pour être utilisé dans un coffret.</li> <li>• Installez les modules dans des conditions de fonctionnement normales.</li> <li>• L'alimentation des capteurs doit uniquement servir à alimenter les capteurs connectés au module.</li> <li>• Utilisez un fusible approuvé CEI60127 sur le circuit d'alimentation et le circuit de sortie pour satisfaire aux exigences de tension et de courant. Fusible recommandé : Littelfuse 5x20 mm à action retardée 218000/Type T.</li> </ul> <p><b>Le non-respect de ces précautions peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.</b></p>



---

## A propos de ce manuel



---

### Présentation

#### Objectif du document

Le manuel de référence du logiciel des automates programmables Twido est composé des sections suivantes :

- Description du logiciel de programmation Twido et introduction aux notions fondamentales requises pour programmer les automates Twido.
- Description des communications, de la gestion des E/S analogiques, de l'installation du module d'interface de bus AS-Interface et d'autres fonctions spéciales.
- Description des langages logiciels utilisés pour créer des programmes Twido.
- Description des instructions et des fonctions des automates Twido.

#### Champ d'application

Les informations du présent manuel s'appliquent **uniquement** aux automates programmables Twido.

**Avertissements  
liés au(x)  
produit(s)**

Schneider Electric ne saurait être tenu responsable des erreurs éventuelles contenues dans ce document. Aucune partie de ce document ne peut être reproduite sous quelque forme que ce soit, ni par aucun moyen que ce soit, y compris électronique, sans la permission écrite préalable de Schneider Electric.

---

**Commentaires  
utilisateur**

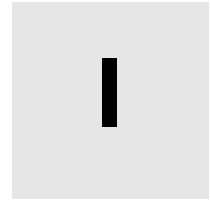
Envoyez vos commentaires à l'adresse e-mail [TECHCOMM@modicon.com](mailto:TECHCOMM@modicon.com)

---



---

# Description du logiciel Twido



---

## Présentation

### Objet de cette partie

Cette rubrique fournit une introduction aux langages du logiciel, ainsi que les principales informations requises pour créer des programmes de régulation des automates programmables Twido.

### Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
1	Introduction au logiciel Twido	19
2	Objets langage Twido	25
3	Mémoire utilisateur	55
4	Modes de fonctionnement de l'automate	67
5	Gestion des tâches événementielles	85



---

# Introduction au logiciel Twido

# 1

---

## Présentation

### Objet de ce chapitre

Ce chapitre offre une introduction rapide à TwidoSoft, le logiciel de programmation et de configuration des automates Twido, ainsi qu'aux langages de programmation Grafcet, liste d'instructions ou schéma à contacts.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Introduction à TwidoSoft	20
Introduction aux langages Twido	21

## Introduction à TwidoSoft

---

### Introduction

TwidoSoft est un environnement de développement graphique permettant de créer, configurer et gérer des applications pour automates programmables Twido. TwidoSoft vous permet de créer des programmes avec différents types de langage (Voir *Langages Twido*, p. 21), puis de transférer l'application en vue de son exécution sur un automate.

### TwidoSoft

TwidoSoft est un programme 32 bits pour PC fonctionnant sous Windows 98 deuxième édition, Windows 2000 Professionnel et Microsoft Windows XP. Principales fonctionnalités logicielles offertes par TwidoSoft :

- interface utilisateur Windows standard
- programmation et configuration d'automates Twido
- connexion et contrôle d'automates

<b>Note :</b> La liaison Automate-PC utilise le protocole TCP/IP. Il est obligatoire que ce protocole soit installé sur le PC.
--

### Configuration minimale

La configuration minimale requise pour l'utilisation de TwidoSoft est :

- Pentium 300MHz,
- 128 Mo de RAM,
- 40 Mo de place disponible sur le disque dur.

## Introduction aux langages Twido

---

### Introduction

Un automate programmable lit des entrées, commande des sorties et résout une logique basée sur un programme. La création d'un programme d'un automate Twido consiste à écrire une série d'instructions rédigées dans un des langages de programmation Twido.

---

### Langages Twido

Les langages suivants peuvent être utilisés pour créer des programmes d'automates Twido :

- Langage liste d'instructions :  
Un programme liste d'instructions est constitué d'une série d'expressions logiques, rédigées sous la forme d'une séquence d'instructions booléennes.
- Langage schéma à contacts :  
Un schéma à contacts est une représentation graphique d'une expression logique.
- Langage Grafcet :  
Le langage grafcet est constitué d'une succession d'étapes et de transitions. Twido comprend les instructions liste Grafcet, mais pas les objets de représentation graphique Grafcet.

Les opérations de création et d'édition de programmes Twido à l'aide de ces langages de programmation peuvent être réalisées depuis un ordinateur personnel (PC).

Une fonctionnalité de réversibilité liste d'instructions / schéma à contacts vous permet de convertir un programme en langage liste d'instructions dans le langage schéma à contacts, et vice-versa.

---

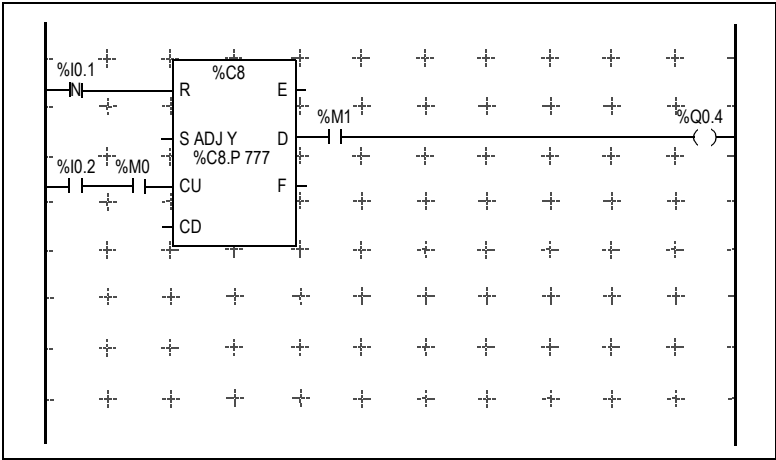
**Langage liste  
d'instructions**

Un programme rédigé en langage liste d'instructions consiste en une série d'instructions exécutées de manière séquentielle par l'automate. Vous trouverez ci-dessous un exemple de programme en langage liste d'instructions.

```
0  BLK  %C8
1  LDF  %I0.1
2  R
3  LD   %I0.2
4  AND  %M0
5  CU
6  OUT_BLK
7  LD   D
8  AND  %M1
9  ST   %Q0.4
10 END_BLK
```

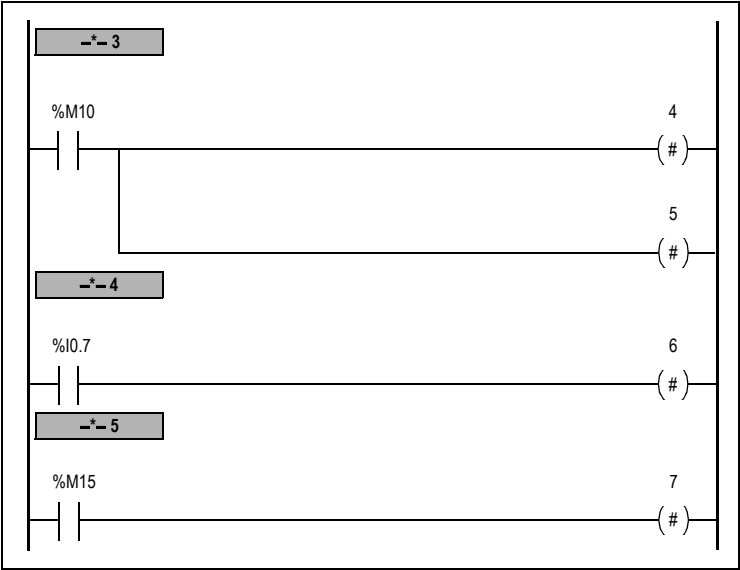
**Langage schéma  
à contacts**

Les schémas à contacts utilisent la même représentation graphique que celle des circuits de relais en logique programmée. Dans ces schémas, les éléments graphiques, tels que des bobines, des contacts et des blocs représentent les instructions du programme. Ci-dessous un exemple de schéma à contacts.



**Langage Grafcet** La méthode analytique Grafcet divise toute application d'automatisation en une série d'étapes auxquelles des actions, des transitions et des conditions sont associées. Vous trouverez ci-dessous des exemples d'instructions Grafcet, rencontrées respectivement dans des programmes liste d'instructions et schéma à contacts.

0	-*-	3
1	LD	%M10
2	#	4
3	#	5
4	-*-	4
5	LD	%I0.7
6	#	6
7	-*-	5
8	LD	%M15
9	#	7
10	...	







---

# Objets langage Twido

## 2

---

### Présentation

#### Objet de ce chapitre

Ce chapitre offre une description détaillée des objets langage de programmation des automates Twido.

#### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Validation d'un objet langage	26
Objets bits	27
Objets mots	30
Objets flottants et mots doubles	34
Adressage d'objets bits	39
Adressage d'objets mots	40
Adressage d'objets flottants	41
Adressage d'objets mots doubles	42
Repérage des entrées/sorties	43
Adressage réseau	45
Objets blocs fonction	46
Objets structurés	48
Objets indexés	52
Symbolisation d'objets	54

## Validation d'un objet langage

---

### Introduction

Les objets mots et bits ne sont valides que lorsqu'ils ont été alloués à une zone mémoire de l'automate. Pour que cette allocation soit possible, il est nécessaire que ces objets aient été utilisés dans l'application avant d'être téléchargés vers l'automate.

### Exemple

La plage d'objets valides est comprise entre 0 et la référence maximum autorisée pour ce type d'objet. Par exemple, si la référence maximum autorisée pour les mots mémoire dans votre application est %MW9, les zones %MW0 à %MW9 sont allouées. Dans cet exemple, %MW10 n'est pas valide. Aucun accès à cette zone n'est autorisé, aussi bien de manière interne qu'externe.

---

## Objets bits

---

### Introduction

Les objets bits sont des variables logicielles de type bit qui peuvent être utilisés comme des opérandes et testés par des instructions booléennes. Vous trouverez ci-dessous la liste des objets bits :

- Bits d'E/S
  - Bits internes (bits mémoire)
  - Bits système
  - Bits étape
  - Bits extraits de mots
-

## Liste des bits opérandes

Le tableau suivant répertorie et décrit les principaux objets bits qui sont utilisés comme opérandes dans des instructions booléennes.

Type	Description	Repère ou valeur	Nombre maximal	Accès en écriture (1)
Valeurs immédiates	0 ou 1 (False ou True)	0 ou 1	-	-
Entrées Sorties	Ces bits sont les "images logiques" des états électriques des E/S. Ils sont stockés dans la mémoire de données et sont mis à jour à chaque scrutation de la logique du programme.	%Ix.y.z (2) %Qx.y.z (2)	Remarque (4)	Non Oui
AS-Interface Entrées Sorties	Ces bits sont les "images logiques" des états électriques des E/S. Ils sont stockés dans la mémoire de données et sont mis à jour à chaque scrutation de la logique du programme.	%IAx.y.z %QAx.y.z	Remarque (5)	Non Oui
Interne (mémoire)	Les bits internes sont des zones de mémoire internes utilisées pour stocker des valeurs intermédiaires lorsqu'un programme est en cours d'exécution. <b>Remarque :</b> Les bits d'E/S non utilisés ne peuvent pas être employés comme des bits internes.	%Mi	128 TWDLC•A10 DRF, TWDLC•A16 DRF 256 Tous les autres automates	Oui
Système	Les bits système %S0 à %S127 surveillent le bon fonctionnement de l'automate ainsi que la bonne exécution du programme de l'application.	%Si	128	Selon i
Blocs fonction	Les bits des blocs fonction correspondent aux sorties des blocs fonction. Ces sorties peuvent être directement câblées ou exploitées en tant qu'objet.	%Tmi.Q, %Ci.P, etc.	Remarque (4)	Non (3)

Type	Description	Repère ou valeur	Nombre maximal	Accès en écriture (1)
Blocs fonction réversibles	Blocs fonction programmés à l'aide d'instructions de programmation réversible BLK, OUT_BLK et END_BLK.	E, D, F, Q, TH0, TH1	Remarque (4)	Non
Extraits de mots	Pour certains mots, un des 16 bits est extrait en tant que bit opérande.	Variable	Variable	Variable
Étapes Grafcet	Les bits %X1 à %Xi sont associés aux étapes Grafcet. Le bit étape Xi est à l'état 1 lorsque l'étape correspondante est active et à l'état 0 lorsqu'elle est désactivée.	%X21	62 TWDLC•A10 DRF, TWDLC•A16 DRF 96 TWDLC•A24 DRF, TWDLCA•40 DRF et automates modulaires	Oui

**Légendes :**

1. Écrit par le programme ou à l'aide de l'éditeur de table d'animation.
2. Reportez-vous à la section "Repérage des Entrées/Sorties".
3. Ces bits, à l'exception de %SBRi.j et de %SCi.j, sont accessibles en écriture et en lecture.
4. Ce nombre est déterminé par le modèle de l'automate.
5. Où, x = adresse du module d'expansion (0..7); y = adresse AS-Interface (0A..31B); z = numéro de voie (0..3). (Voir *Adressage des entrées/sorties associées aux équipements esclaves connectés sur bus AS-Interface V2*, p. 237.)

## Objets mots

---

### Introduction

Les objets mots sont repérés sous la forme de mots de 16 bits rangés dans la mémoire de données et pouvant contenir un entier compris entre -32 768 et 32 767 (sauf pour le bloc fonction compteur rapide (FC) qui est compris entre 0 et 65 535).

Exemples d'objets mots :

- Valeurs immédiates
  - Mots internes (%MWi) (mots mémoire)
  - Mots constants (%KWi)
  - Mots échanges E/S (%IWi, %QWi%)
  - Mots d'E/S analogiques AS-Interface (IWAi, %QWAI)
  - Mots système (%SWi)
  - Blocs fonction (données de configuration et/ou d'exécution)
- 

### Formats de mot

Le contenu des mots ou des valeurs est rangé dans la mémoire utilisateur sous la forme d'un code binaire à 16 bits (complément à deux) utilisant la convention suivante :

Position du bit																
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
0	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1	Etat du bit
																Valeur du bit
+	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	

Pour les notations binaires signées, le bit 15 est attribué, par convention, au signe de la valeur codée :

- Le bit 15 est réglé sur 0 : le mot contient une valeur positive.
- Le bit 15 est réglé sur 1 : le mot contient une valeur négative (les valeurs négatives sont exprimées en complément de deux).

Il est possible d'entrer et de récupérer les mots et les valeurs immédiates sous les formats suivants :

- Décimal  
Min : -32 768, Max : 32 767 (1 579, par exemple)
  - Hexadécimal  
Min : 16#0000, Max : 16#FFFF (16#A536, par exemple)  
Syntaxe alternative : #A536
-

**Description des objets mots**

Le tableau suivant décrit les objets mots.

Mots	Description	Repère ou valeur	Nombre maximal	Accès en écriture (1)
Valeurs immédiates	Il s'agit d'entiers dont le format est identique à celui des mots de 16 bits. Cela permet d'attribuer des valeurs à ces mots.		-	Non
	Base 10	-32 768 à 32 767		
	Base 16	16#0000 à 16#FFFF		
Interne (mémoire)	Mots utilisés pour ranger des valeurs dans la mémoire des données au cours du fonctionnement. Les mots %MW0 à %MW255 sont directement lus et écrits par le programme.	%MWi	3 000	Oui
Constante	Mémoirisent les constantes ou les messages alphanumériques. Leur contenu peut être écrit ou modifié uniquement à l'aide de TwidoSoft et en cours de configuration. Le programme ne peut accéder aux mots constants allant de %KW0 à %KW63 qu'en lecture.	%KWi	256	Oui, uniquement à l'aide de TwidoSoft
Système	Ces mots de 16 bits comportent plusieurs fonctions : <ul style="list-style-type: none"> <li>• Ils permettent l'accès aux données provenant directement de l'automate en lisant les mots %SWi.</li> <li>• Ils effectuent des opérations sur l'application (l'ajustement des blocs horodateurs, par exemple).</li> </ul>	%SWi	128	Selon i
Blocs fonction	Ces mots correspondent aux paramètres ou aux valeurs courantes des blocs fonction.	%TM2.P, %Ci.P, etc.		Oui

Mots	Description	Repère ou valeur	Nombre maximal	Accès en écriture (1)
Mots d'échange réseau	Attribués aux automates connectés en tant que Liaisons distantes. Ces mots sont utilisés pour la communication entre les automates :			
	Entrée réseau	%INWi.j	4 par liaison distante	Non
	Sortie réseau	%QNWj.j	4 par liaison distante	Oui
Mots d'E/S analogiques	Attribués aux entrées et sorties analogiques des modules esclaves AS-Interface.			
	Entrées analogiques	%IWAx.y.z	Remarque (3)	Non
	Sorties analogiques	%QWAx.y.z	Remarque (3)	Oui
Bits extraits	Il est possible d'extraire un des 16 bits à partir des mots suivants :			
	Circuit interne	%MWi:Xk	1 500	Oui
	Système	%SWi:Xk	128	Dépend de i
	Constante	%KWj:Xk	64	Non
	Entrée	%IWj.j:Xk	Remarque (2)	Non
	Sortie	%QWj.j:Xk	Remarque (2)	Oui
	Entrée esclave AS-Interface	%IWAx.y.z:Xk	Remarque (2)	Non
	Sortie esclave AS-Interface	%QWAx.y.z:Xk	Remarque (2)	Oui
	Entrée réseau	%INWi.j:Xk	Remarque (2)	Non
	Sortie réseau	%QNWj.j:Xk	Remarque (2)	Oui



**Note :**

1. Ecrit par le programme ou à l'aide de l'éditeur de table d'animation.
  2. Ce nombre est déterminé par la configuration.
  3. Où, x = adresse du module d'expansion (0..7); y = adresse AS-Interface (0A..31B); z = numéro de voie (0..3). (Voir *Adressage des entrées/sorties associées aux équipements esclaves connectés sur bus AS-Interface V2*, p. 237.)
-

## Objets flottants et mots doubles

---

### Introduction

Le logiciel TwidoSoft permet d'effectuer des opérations sur les objets flottants et mots doubles entiers.

Un flottant est un argument mathématique qui possède une décimale dans son expression (exemples : 3,4E+38, 2,3 ou 1,0).

Un mot double entier est constitué de 4 octets stockés dans la mémoire de données et contenant une valeur comprise entre -2 147 483 648 et +2 147 483 647.

---

**Format et valeur du flottant**

Le format flottant utilisé est celui de la norme IEEE STD 734-1985 (équivalence CEI 559). La longueur des mots est de 32 bits, ce qui correspond à des nombres flottants simple précision.

Tableau représentant le format d'un flottant :

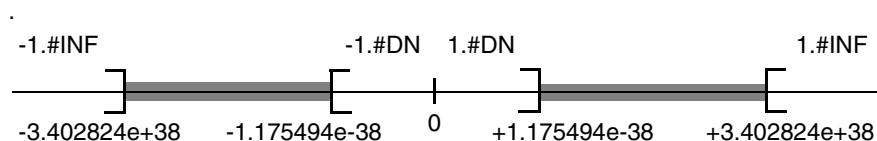
Bit 31	Bits {30...23}	Bits {22...0}
S	Exposant	Mantisse

La valeur du format ci-dessus est déterminée par l'équation suivante :

$$\text{Valeur Flottant 32 bits} = (-1)^S * 2^{(\text{Exposant} - 127)} * 1, \text{Mantisse}$$

Les valeurs flottantes peuvent être représentées avec ou sans exposant, elles doivent toujours comporter une virgule (virgule flottante).

Les valeurs flottantes sont comprises entre -3,402824e+38 à -1,175494e-38 et 1,175494e-38 à 3,402824e+38 (valeurs grisées sur le schéma). Elles comportent aussi la valeur 0 notée 0,0.



Lorsqu'un résultat de calcul est :

- inférieur à -3,402824e+38, le symbole -1.#INF (pour -infini) est affiché.
- supérieur à +3,402824e+38, le symbole 1.#INF (pour + infini) est affiché.
- compris entre -1,175494e-38 et 1,175494e-38, il est arrondi à 0,0. Une valeur comprise entre ces bornes ne peut être saisie en valeur flottante.
- indéfini (par exemple, racine carrée d'un nombre négatif), le symbole 1.#NAN ou -1.#NAN est affiché.

La précision de la représentation est de 2-24. Pour la visualisation des nombres flottants, il est inutile d'afficher plus de 6 chiffres après la virgule.

**Note :**

- La valeur "1 285" est interprétée en tant que valeur entière. Pour pouvoir être prise en compte comme valeur flottante, elle doit être écrite sous la forme suivante : "1 285,0"

Plage limite des fonctions arithmétiques sur des objets flottants

Le tableau suivant décrit la plage limite des fonctions arithmétiques sur des objets flottants :

Fonction arithmétique		Plage limite et opérations invalides	
Type	Syntaxe	#QNAN (Non valide)	#INF (Infini)
Racine carrée d'un opérande	SQRT(x)	x < 0	x > 1,7E38
Alimentation d'un entier par un réel EXPT(%MF,%MW)	EXPT(y, x) (où : x^y = %MW^%MF)	x < 0	y.ln(x) > 88
Logarithme de base 10	LOG(x)	x <= 0	x > 2,4E38
Logarithme naturel	LN(x)	x <= 0	x > 1,65E38
Exponentiel naturel	EXP(x)	x < 0	x > 88.0

Compatibilité matérielle

Les opérations sur flottants et mots doubles ne sont pas prises en charge par tous les automates Twido.

Le tableau suivant décrit la compatibilité matérielle :

Automate Twido	Mots doubles pris en charge	Flottants pris en charge
TWDLMDA40DUK	Oui	Oui
TWDLMDA40DTK	Oui	Oui
TWDLMDA20DUK	Oui	Non
TWDLMDA20DTK	Oui	Non
TWDLMDA20DRT	Oui	Oui
TWDLCA•40DRF	Oui	Oui
TWDLCA•A24DRF	Oui	Non
TWDLCA•A16DRF	Oui	Non
TWDLCA•A10DRF	Non	Non

**Contrôle de validité**

Le bit système %S18 est mis à 1 lorsque le résultat ne se situe pas dans la plage valide.

Les bits de mot d'état %SW17 indiquent la cause d'une erreur au niveau d'une opération sur valeur flottante.

Différents bits du mot %SW17 :

%SW17:X0	Opération incorrecte, le résultat n'est pas un nombre (1.#NAN ou -1.#NAN)
%SW17:X1	Réservé
%SW17:X2	Division par 0, le résultat est l'infini (-1.#INF ou 1.#INF)
%SW17:X3	Résultat supérieur en valeur absolue à +3,402824e+38, le résultat est l'infini (-1.#INF ou 1.#INF)
%SW17:X4 à X15	Réservé

Ce mot est remis à 0 par le système lors d'un démarrage à froid et par le programme pour une réutilisation.

**Description des objets flottants et mots doubles**

Le tableau suivant décrit les objets flottants et mots doubles :

Type d'objet	Description	Repère	Nombre maximal	Accès en écriture	Forme indexée
Valeurs immédiates	Entiers ou décimaux dont le format est identique à des objets de 32 bits.	-	[-]	Non	-
Flottant interne	Objets utilisés pour stocker des valeurs dans la mémoire des données lorsque le système est en cours d'exécution.	%MFi	1500	Oui	%MFi[index]
Mot double interne		%MDi	1500	Oui	%MDi[index]
Constante flottante	Mémorise les constantes.	%KFi	128	Oui, uniquement à l'aide de TwidoSoft	%KFi[index]
Constante double		%KDi	128	Oui, uniquement à l'aide de TwidoSoft	%KDi[index]

**Possibilité de recouvrement entre objets**

Les mots longueur simple, double et flottants sont stockés au sein de l'espace des données dans une même zone mémoire. Ainsi, le mot flottant %MFi et le mot double %MDi correspondent aux mots longueur simple %MWi et %MWi+1 (le mot %MWi contenant les bits de poids faible et le mot %MWi+1 les bits de poids fort du mot %MFi).

Le tableau suivant illustre le recouvrement des mots flottants et des mots doubles internes :

Flottant et double	Repère impair	Mots internes
%MF0 / %MD0		%MW0
	%MF1 / %MD1	%MW1
%MF2 / %MD2		%MW2
	%MF3 / %MD3	%MW3
%MF4 / %MD4		%MW4
	...	%MW5
...		...
	%MFi / %MDi	%MWi
%MFi+1 / %MDi+1		%MWi+1

Le tableau suivant illustre le recouvrement des constantes flottantes et doubles :

Flottant et double	Repère impair	Mots internes
%KF0 / %KD0		%KW0
	%KF1 / %KD1	%KW1
%KF2 / %KD2		%KW2
	%KF3 / %KD3	%KW3
%KF4 / %KD4		%KW4
	...	%KW5
...		...
	%kFi / %kDi	%KW <sub>i</sub>
%KFi+1 / %KDi+1		%KW <sub>i</sub> +1

**Exemple :**

%MF0 correspond à %MW0 et %MW1. %KF543 correspond à %KW543 et %KW544.

---

## Adressage d'objets bits

### Syntaxe

L'adressage des objets bits d'étape, internes et système doit se conformer à la syntaxe suivante :

%	M, S ou X	i
Symbole	Type d'objet	Numéro

### Description

Le tableau suivant décrit les éléments de la syntaxe d'adressage.

Groupe	Elément	Description
Symbole	%	Une variable logicielle doit toujours débuter par un symbole de pourcentage (%).
Type d'objet	M	Les bits internes permettent de stocker des valeurs intermédiaires lorsqu'un programme est en cours d'exécution.
	S	Les bits système donnent des informations d'état et de contrôle relatives à l'automate.
	X	Les bits d'étape offrent des informations sur l'état des activités des étapes.
Numéro	i	La valeur maximum dépend du nombre d'objets configurés.

### Exemples d'adressage d'objets bits :

- %M25 = bit interne numéro 25
- %S20 = bit système numéro 20
- %X6 = bit étape numéro 6

### Objets bits extraits de mots

TwidoSoft permet d'extraire un des 16 bits des mots. L'adresse du mot est alors complétée par le rang du bit extrait suivant la syntaxe suivante :

MOT	X	k
Adresse du mot		Position k = 0 - 15 rang du bit dans l'adresse du mot.

### Exemples :

- %MW5:X6 = bit numéro 6 du mot interne %MW5
- %QW5.1:X10 = bit numéro 10 du mot de sortie %QW5.1

## Adressage d'objets mots

### Introduction

L'adressage d'objets mots doit se conformer à la syntaxe décrite ci-dessous. Veuillez noter que cette syntaxe ne s'applique pas à l'adressage d'E/S (reportez-vous à la rubrique *Repérage des entrées/sorties*, p. 43) et des blocs fonctions (reportez-vous à la rubrique *Objets blocs fonction*, p. 46).

### Syntaxe

L'adressage des mots internes, constants et système doit se conformer à la syntaxe suivante :

%	M, K ou S	W	i
Symbole	Type d'objet	Syntaxe	Numéro

### Description

Le tableau suivant décrit les éléments de la syntaxe d'adressage.

Groupe	Élément	Description
Symbole	%	Une adresse interne doit toujours débuter par un symbole de pourcentage (%).
Type d'objet	M	Les mots internes permettent de stocker des valeurs intermédiaires lorsqu'un programme est en cours d'exécution.
	K	Les mots constants permettent de stocker des valeurs constantes ou des messages alphanumériques. Leur contenu ne peut être écrit ou modifié qu'en utilisant TwidoSoft.
	S	Les mots système offrent des informations d'état et de régulation relatives à l'automate.
Syntaxe	W	Mot de 16 bits.
Numéro	i	La valeur maximum dépend du nombre d'objets configurés.

### Exemples d'adressage d'objets mots :

- %MW15 = mot interne numéro 15
- %KW26 = mot constant numéro 26
- %SW30 = mot système numéro 30



## Adressage d'objets flottants

**Introduction** L'adressage d'objets flottants doit se conformer à la syntaxe décrite ci-dessous. Veuillez noter que cette syntaxe ne s'applique pas à l'adressage d'E/S (reportez-vous à la rubrique *Repérage des entrées/sorties*, p. 43) et des blocs fonctions (reportez-vous à la rubrique *Objets blocs fonction*, p. 46).

**Syntaxe** L'adressage des flottants internes et constants doit se conformer à la syntaxe suivante :

%	M ou K	F	i
Symbole	Type d'objet	Syntaxe	Numéro

**Description** Le tableau suivant décrit les éléments de la syntaxe d'adressage.

Groupe	Elément	Description
Symbole	%	Une adresse interne doit toujours débuter par un symbole de pourcentage (%).
Type d'objet	M	Les flottants internes permettent de stocker des valeurs intermédiaires lorsqu'un programme est en cours d'exécution.
	K	Les flottants constants permettent de stocker des valeurs constantes. Leur contenu ne peut être écrit ou modifié qu'en utilisant TwidoSoft.
Syntaxe	F	Objet de 32 bits.
Numéro	i	La valeur maximum dépend du nombre d'objets configurés.

**Exemples d'adressage d'objets flottants :**

- %MF15 = flottant interne numéro 15
- %KF26 = flottant constant numéro 26

## Adressage d'objets mots doubles

**Introduction** L'adressage des objets mots doubles doit se conformer à la syntaxe décrite ci-dessous. Veuillez noter que cette syntaxe ne s'applique pas à l'adressage d'E/S (reportez-vous à la rubrique *Repérage des entrées/sorties, p. 43*) et des blocs fonctions (reportez-vous à la rubrique *Objets blocs fonction, p. 46*).

**Syntaxe** L'adressage des mots doubles internes et constants doit se conformer à la syntaxe suivante :

%	M ou K	D	i
Symbole	Type d'objet	Syntaxe	Numéro

**Description** Le tableau suivant décrit les éléments de la syntaxe d'adressage.

Groupe	Élément	Description
Symbole	%	Une adresse interne doit toujours débuter par un symbole de pourcentage (%).
Type d'objet	M	Les mots doubles internes permettent de stocker des valeurs intermédiaires lorsqu'un programme est en cours d'exécution.
	K	Les mots doubles constants permettent de stocker des valeurs constantes ou des messages alphanumériques. Leur contenu ne peut être écrit ou modifié qu'en utilisant TwidoSoft.
Syntaxe	D	Double mot de 32 bits.
Numéro	i	La valeur maximum dépend du nombre d'objets configurés.

**Exemples d'adressage d'objets mots doubles :**

- %MD15 = mot double interne numéro 15
- %KD26 = mot double constant numéro 26

## Repérage des entrées/sorties

### Introduction


Chaque point d'E/S (entrée/sortie) d'une configuration Twido possède un repère unique. Par exemple, le repère "%I0.0.4" ne représente que l'entrée 4 d'un automate. Des repères d'E/S peuvent être affectés aux matériels suivants :

- Automate configuré en tant que maître de liaison distante
- Automate configuré en tant qu'E/S distante
- Modules d'E/S d'expansion

Le module interface bus AS-Interface TWDNOI10M3 dispose d'un adressage spécifique des entrées/sorties de ses équipements esclaves (Voir *Adressage des entrées/sorties associées aux équipements esclaves connectés sur bus AS-Interface V2*, p. 237).

### Références multiples à une sortie ou à une bobine

Un programme peut comporter plusieurs références à une même sortie ou bobine. Seul le résultat de la dernière référence traitée est mis à jour au niveau des sorties du matériel. Par exemple, %Q0.0.0 peut être utilisé plusieurs fois dans un programme sans qu'un avertissement ne signale la multiplicité des occurrences. Il est donc important de ne valider que l'équation qui donnera l'état souhaité de la sortie.

	<b>ATTENTION</b>
	<b>Opération inattendue</b>
	Les doublons de sortie ne sont pas contrôlés et aucun avertissement n'est donné. Vérifiez l'utilisation qui est faite des sorties et des bobines avant de les modifier dans l'application.  <b>Le non-respect de ces précautions peut entraîner des lésions corporelles ou des dommages matériels.</b>

### Format

Le repérage des entrées et des sorties doit se conformer à la syntaxe suivante.

%	I, Q	x	.	y	.	z
Symbole	Type d'objet	Position de l'automate	point	Type d'E/S	point	Numéro de voie

Le repérage des mots d'échange en entrée et en sortie doit se conformer à la syntaxe suivante.

%	I, Q	W	x	.	y
Symbole	Type d'objet	Format	Position de l'automate	point	Type d'E/S

**Description**

Le tableau suivant décrit la syntaxe de repérage des E/S.

Groupe	Elément	Valeur	Description
Symbole	%	-	Un repère interne doit toujours débiter par un symbole de pourcentage (%).
Type d'objet	I	-	Entrée. "Image logique" de l'état électrique de l'entrée d'un automate ou d'un module d'E/S d'expansion.
	Q	-	Sortie. "Image logique" de l'état électrique de la sortie d'un automate ou d'un module d'E/S d'expansion.
Position de l'automate	x	0 1 - 7	Automate maître (maître de liaison distante). Automate distant (esclave de liaison distante).
Type d'E/S	y	0 1 - 7	E/S de base (E/S locale sur un automate). Modules d'E/S d'expansion.
Numéro de voie	z	0 - 31	Numéro de la voie d'E/S sur l'automate ou le module d'E/S d'expansion. Le nombre de points d'E/S disponibles dépend du modèle de l'automate ou du type du module d'E/S d'expansion.

**Exemples**

Le tableau suivant présente quelques exemples de repérage des E/S.

Objet d'E/S	Description
%I0.0.5	Entrée n° 5 sur la base automate (E/S locale).
%Q0.3.4	Sortie n° 4 sur le module d'E/S d'expansion d'adresse 3 pour la base automate (E/S d'expansion).
%I0.0.3	Entrée n° 3 sur la base automate.
%I3.0.1	Entrée n° 1 sur l'automate d'E/S distant d'adresse 3 de la liaison distante.
%I0.3.2	Entrée n° 2 sur le module d'E/S d'expansion d'adresse 3 pour la base automate.

## Adressage réseau

**Introduction** Les mots réseau %INW et %QNW permettent d'échanger des données d'application entre les automates d'extension et l'automate maître sur un réseau de liaison distante Twido. Reportez-vous au chapitre *Communications* , p. 93 pour obtenir plus d'informations.

**Format** L'adressage réseau doit se conformer à la syntaxe suivante.

%	IN,QN	W	x	.	j
Symbole	Type d'objet	Format	Position de l'automate	point	Mot

**Description de la syntaxe** Le tableau suivant décrit la syntaxe d'adressage réseau.

Groupe	Elément	Valeur	Description
Symbole	%	-	Un repère interne doit toujours débuter par un symbole de pourcentage (%).
Type d'objet	IN	-	Mot d'entrée réseau. Transfert de données de l'automate maître vers l'automate d'extension.
	QN	-	Mot de sortie réseau. Transfert de données de l'automate d'extension vers l'automate maître.
Format	W	-	Mot de 16 bit.
Position de l'automate	x	0	Automate maître (maître de liaison distante).
		1 - 7	Automate distant (esclave de liaison distante).
Mot	j	0 - 3	Chaque automate d'extension utilise un maximum de quatre mots pour assurer l'échange de données avec l'automate maître.

**Exemples** Le tableau suivant présente quelques exemples d'adressage réseau.

Objet réseau	Description
%INW3.1	Mot réseau n°1 de l'automate distant n°3.
%QNW0.3	Mot réseau n°3 de la base automate.

## Objets blocs fonction

---

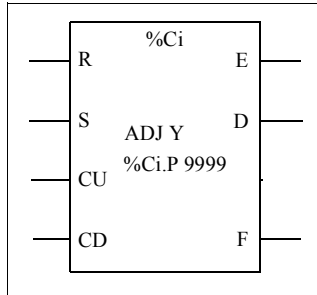
### Introduction

Les blocs fonction contiennent des objets bits et des mots spécifiques accessibles par le programme.

---

### Exemple de bloc fonction

L'illustration suivante présente un bloc fonction compteur.



Bloc compteur/décompteur

---

### Objets bits

Les objets bits correspondent aux sorties des blocs. Les instructions booléennes de test permettent d'accéder à ces bits selon l'une ou l'autre de ces méthodes :

- directement (LD E, par exemple) si les bits sont reliés au bloc par une programmation réversible (voir *Principes de programmation de blocs fonction standards*, p. 335) ;
- en spécifiant le type de bloc (LD %Ci.E, par exemple).

Les instructions permettent d'accéder aux entrées.

---

### Objets mots

Les objets mots correspondent aux paramètres et valeurs spécifiés suivants :

- Paramètres de configuration des blocs : le programme peut accéder à certains paramètres (paramètres de présélection, par exemple), mais pas à d'autres (base temps, par exemple).
  - Valeurs courantes : %Ci.V, la valeur de comptage courante, par exemple.
-

**Objets mots**

Les objets mots doubles augmentent les capacités de calcul de votre automate Twido lors de l'exécution de fonctions système telles que les compteurs rapides (%FC ou %VFC) et les générateurs d'impulsions (%PLS).

Le repérage des objets mots doubles 32 bits utilisés avec les blocs fonction consiste uniquement à ajouter des objets mots standard avec le caractère "D" à la syntaxe d'origine. L'exemple suivant indique comment repérer la valeur courante d'un compteur rapide (FC) au format standard et au format mot double.

- %FCi.V est la valeur courante du compteur rapide (FC) au format standard.
- %FCi.VD est la valeur courante du compteur rapide (FC) au format mot double.

**Note :** Les objets mots doubles ne sont pas pris en charge par tous les automates Twido. Reportez-vous au sous-chapitre *Compatibilité matérielle*, p. 36 pour savoir si votre automate Twido accepte les mots doubles.

---

**Objets  
accessibles par  
le programme**

Reportez-vous aux sous-chapitres suivants pour connaître la liste des objets accessibles par le programme.

- Pour les blocs fonction élémentaires, reportez-vous au sous-chapitre *Blocs fonctions standards*, p. 333.
  - Pour les blocs fonction avancés, reportez-vous au sous-chapitre *Objets mots et objets bits associés à des blocs fonction avancés*, p. 388.
-

## Objets structurés

---

### Introduction

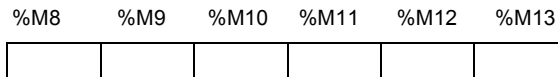
Les objets structurés sont des ensembles formés par des objets adjacents. Twido prend en charge les types d'objet structuré suivants :

- Chaînes de bits
  - Tables de mots
  - Tables de mots doubles
  - Tables de mots flottants
- 

### Chaînes de bits

Les chaînes de bits sont composées d'une série de bits objet adjacent du même type et dont la longueur (L) est définie.

**Exemple :** Chaîne de bits %M8:6



<b>Note :</b> %M8:6 est correct (car 8 est un multiple de 8), alors que %M10:16 ne l'est pas (10 n'est pas un multiple de 8).
---

Les chaînes de bits peuvent être utilisées avec l'instruction d'affectation (voir *Instructions d'affectation*, p. 360).

---



Types de bit disponibles

Types de bit disponibles pour les chaînes de bits :

Type	Repère	Taille maximale	Accès en écriture
Bits d'entrée TOR	%I0.0:L ou %I1.0:L (1)	0<L<17	Non
Bits de sortie TOR	%Q0.0:L ou %Q1.0:L (1)	0<L<17	Oui
Bits système	%Si:L où "i" est multiple de 8	0<L<17 et i+L≤ 128	En fonction de i
Bits pas Grafcet	%Xi:L où "i" est multiple de 8	0<L<17 et i+L≤ 95 (2)	Oui (via le programme)
Bits internes	%Mi:L où "i" est multiple de 8	0<L<17 et i+L≤ 256 (3)	Oui

Légende :

1. Seuls les bits d'E/S 0 à 16 peuvent être lus en chaîne de bits. Pour les automates à 24 entrées et les modules à 32 E/S, les bits supérieurs à 16 ne peuvent pas être lus en chaîne de bits.
2. Le maximum de i+L pour les automates TWWDLCAA10DRF et TWDLCAA16DRF est 62.
3. Le maximum de i+L pour les automates TWWDLCAA10DRF et TWDLCAA16DRF est 128.

Tables de mots

Les tables de mots sont composées d'une série d'objets adjacents du même type et dont la longueur (L) est définie.

**Exemple :** Table de mots %KW10:7

%KW10	16 bits
%KW16	

Les tables de mots peuvent être utilisées avec l'instruction d'affectation (voir *Instructions d'affectation, p. 360*).

Types de mot disponibles

Types de mot disponibles pour les tables de mots :

Type	Repère	Taille maximale	Accès en écriture
Mots internes	%MWi:L	0<L<256 et i+L< 3 000	Oui
Mots constante	%KWi:L	0<L<256 et i+L< 256	Non
Mots système	%SWi:L	0<L et i+L<128	En fonction de i

Tables de mots doubles

Les tables de mots doubles sont composées d'une série d'objets adjacents du même type et dont la longueur (L) est définie.

**Exemple :** Table de mots doubles %KD10:7

%KD10	32 bits
%KD22	

Les tables de mots doubles peuvent être utilisées avec l'instruction d'affectation (voir *Instructions d'affectation*, p. 360).

Types de mot double disponibles

Types de mot disponibles pour les tables de mots doubles :

Type	Repère	Taille maximale	Accès en écriture
Mots internes	%MDi:L	0<L<256 et i+L< 3 000	Oui
Mots constante	%KDi:L	0<L et i+L< 256	Non

**Tables de mots flottants**

Les tables de mots flottants sont composées d'une série d'objets adjacents du même type et dont la longueur (L) est définie.  
**Exemple :** Table de mots flottants %KF10:7

%KF10	32 bits
%KF22	

Les tables de mots flottants peuvent être utilisées avec l'instruction d'affectation (voir Instructions d'affectation).

**Types de mot flottant disponibles**

Types de mot disponibles pour les tables de mots flottants :

Type	Repère	Taille maximale	Accès en écriture
Mots internes	%MFi:L	0<L<256 et i+L< 3 000	Oui
Mots constante	%KFi:L	0<L et i+L<256	Non

## Objets indexés

---

### Introduction

Un mot indexé est un mot simple ou double ou un flottant comportant un repère d'objet indexé. Il existe deux types de repérage d'objet :

- repérage direct
- repérage indexé

### Repérage direct

Le repère direct d'un objet est défini au moment de l'écriture du programme.

**Exemple :** %M26 est un bit interne dont le repère direct est 26.

### Repérage indexé

L'indexation du repère d'un objet permet de modifier ce repère en attribuant un index au repère direct d'un objet. Le contenu de l'index est ajouté au repère direct de l'objet. L'index est défini par un mot interne %MWi. Le nombre de "mots indexés" est illimité.

**Exemple :** %MW108[%MW2] est un mot dont le repère est composé du repère direct 108 et du contenu du mot %MW2.

Si la valeur du mot %MW2 est 12, le fait d'écrire dans %MW108[%MW2] équivaut à écrire dans %MW120 (108 + 12).

### Objets disponibles pour le repérage indexé

Le tableau suivant répertorie les différents types d'objet disponibles pour le repérage indexé.

Type	Repère	Taille maximale	Accès en écriture
Mots internes	%MWi[%MWj]	0 ≤ i+%MWj < 3000	Oui
Mots constante	%KWi[%MWj]	0 ≤ i+%MWj < 256	Non
Mots doubles internes	%MDi[%MWj]	0 ≤ i+%MWj < 2999	Oui
Mots doubles constante	%KDi[%MWj]	0 ≤ i+%MWj < 255	Non
Flottants internes	%MFi[%MWj]	0 ≤ i+%MWj < 2999	Oui
Flottants constante	%KFi[%MWj]	0 ≤ i+%MWj < 255	Non

Les objets indexés peuvent être utilisés avec les instructions d'affectation (voir *Instructions d'affectation*, p. 360 pour mots simples et doubles) et dans les instructions de comparaison (voir *Instructions de comparaison*, p. 365 pour mots simples et doubles). Ce type de repérage permet de scruter individuellement un ensemble d'objets du même type (tels que des mots internes ou des constantes), en modifiant le contenu de l'objet indexé via le programme.

---

**Bit système de débordement d'index %S20**

Un débordement d'index se produit lorsque le repère d'un objet indexé dépasse les limites de la zone mémoire contenant le même type d'objet. Pour résumer :

- Le repère de l'objet plus le contenu de l'index sont inférieurs à 0.
- Le repère de l'objet plus le contenu de l'index sont supérieurs au plus grand mot directement référencé dans l'application. Le nombre maximum est 2 999 (pour les mots %MWi) ou 255 (pour les mots %KWi).

En cas de débordement d'index, le système provoque la mise à 1 du bit système %S20 et une valeur d'index égale à 0 est affectée à l'objet.

**Note** : L'utilisateur est responsable du contrôle des débordements. Le bit %S20 doit être lu par le programme utilisateur pour un traitement éventuel. La remise à zéro est à la charge de l'utilisateur.

%S20 (état initial = 0) :

- Sur débordement d'index : mise à 1 par le système.
  - Acquiescement de débordement : mise à 0 par l'utilisateur, après modification de l'index.
-

## Symbolisation d'objets

---

### Introduction

Les symboles permettent de repérer des objets du langage logiciel Twido, à l'aide de noms ou de mnémoniques personnalisés. L'utilisation de symboles permet d'examiner et d'analyser rapidement la logique d'un programme et simplifie significativement les procédures de développement et de test d'une application.

### Exemple

Par exemple, le symbole WASH\_END pourrait être utilisé pour identifier un bloc fonction horodateur correspondant à la fin d'un cycle de lavage. L'utilisation de ce nom se révélera beaucoup plus pratique que celui du repère du programme, tel que %TM3.

### Instructions pour la définition de symboles

Les noms de symboles doivent répondre aux exigences suivantes :

- Ces noms doivent comporter un maximum de 32 caractères.
- Ces noms peuvent uniquement comporter des lettres (A-Z), des nombres (0 -9) et des traits de soulignement (\_).
- Le premier caractère de ces noms doit être alphanumérique ou accentué. Ces noms ne peuvent pas comporter de signe de pourcentage (%).
- Ces noms ne peuvent pas contenir d'espaces ou de caractères spéciaux.
- Aucune distinction ne sera faite entre les majuscules et les minuscules. Par exemple, "Pompe1" et "POMPE1" correspondront au même symbole et ne pourront par conséquent être utilisés qu'une seule fois dans l'application.

### Edition des symboles

Utilisez l'éditeur de symboles pour définir et associer des objets de langage. Il est important de signaler que les symboles et leurs commentaires ne sont pas stockés sur l'automate, mais avec l'application, sur le disque dur. Il est donc impossible de transférer ces symboles vers l'automate, avec l'application.

---

# Mémoire utilisateur



---

## Présentation

**Objet de ce chapitre** Ce chapitre offre une description de la structure de la mémoire utilisateur Twido, ainsi que des informations sur son utilisation.

**Contenu de ce chapitre** Ce chapitre contient les sujets suivants :

Sujet	Page
Structure de la mémoire utilisateur	56
Sauvegarde et restauration sans cartouche de sauvegarde, ni cartouche de mémoire étendue	59
Sauvegarde et restauration avec une cartouche de sauvegarde de 32 K	61
Utilisation de la cartouche de mémoire étendue de 64 K	64

---

## Structure de la mémoire utilisateur

---

### Introduction

La mémoire de l'automate accessible par votre application est divisée en deux ensembles distincts :

- les valeurs de bits ;
- les valeurs de mots (valeurs signées à 16 bits) et les valeurs de mots doubles (valeurs signées à 32 bits).

---

### Mémoire bits

La mémoire bits est située dans la mémoire RAM intégrée de l'automate. Elle contient l'image des 128 objets bits.

---

### Mémoire mots

La mémoire mots (16 bits) prend en charge les éléments suivants :

- **Mots dynamiques** : mémoire d'exécution (stockée uniquement dans la RAM).
- **Mots (%MW) et mots doubles (%MD) mémoire** : données dynamiques système et données système.
- **Programme** : descripteurs et code exécutable des tâches.
- **Données de configuration** : mots constante, valeurs initiales et configuration des entrées/sorties.

---

### Types de stockage mémoire

Les automates Twido disposent des trois types de stockage mémoire suivant :

- **RAM**  
Mémoire volatile interne : contient des mots dynamiques, des mots mémoire, des données de configuration et de programme.
  - **EEPROM**  
Mémoire EEPROM intégrée de 32 Ko permettant une sauvegarde interne des données et du programme. Elle protège le programme des altérations causées par une défaillance de pile ou une coupure secteur de plus de 30 jours. Elle contient des données de programme et de configuration. Elle comporte un maximum de 512 mots mémoire. Le programme n'est pas sauvegardé si une cartouche de mémoire étendue de 64 K est en cours d'utilisation et que Twido a été configuré pour accepter cette cartouche de mémoire.
  - **Cartouche de sauvegarde de 32 K**  
Cartouche externe en option utilisée pour enregistrer un programme et transférer ce programme vers d'autres automates Twido. Elle peut être utilisée pour mettre à jour le programme dans la RAM de l'automate. Elle contient un programme et des constantes, mais aucun mot mémoire.
  - **Cartouche de mémoire étendue de 64 K**  
Cartouche externe en option qui stocke un programme jusqu'à 64 K. Doit rester raccordée à l'automate tant que le programme est utilisé.
-



**Enregistrement  
de la mémoire**

Les mots mémoire et le programme de votre automate peuvent être enregistrés dans les éléments suivants :

- RAM (jusqu'à 30 jours avec une pile satisfaisante)
- EEPROM (32 Ko maximum)

Le transfert du programme depuis la mémoire EEPROM vers la mémoire RAM s'effectue automatiquement, lorsqu'il y a perte du programme dans la RAM (ou en cas d'absence de pile).

Notez qu'il est également possible d'effectuer un transfert manuel à l'aide de TwidoSoft.

---

## Configurations de la mémoire

Les tableaux suivants présentent les configurations de mémoire possibles des automates Twido (compacts et modulaires).

Type de mémoire	Automates compacts				
	10DRF	16DRF	24DRF	40DRF (32 k)	40DRF** (64 k)
RAM interne Mém 1*	10 Ko	10 Ko	10 Ko	10 Ko	10 Ko
RAM externe Mém 2*		16 Ko	32 Ko	32 Ko	64 Ko
EEPROM interne	8 Ko	16 Ko	32 Ko	32 Ko	32 Ko***
EEPROM externe	32 Ko	32 Ko	32 Ko	32 Ko	64 Ko
Taille maximale du programme	8 Ko	16 Ko	32 Ko	32 Ko	64 Ko
Sauvegarde externe maximale	8 Ko	16 Ko	32 Ko	32 Ko	64 Ko

Type de mémoire	Automates modulaires		
	20DUK 20DTK	20DRT 40DUK 40DTK (32 k)	20DRT 40DUK 40DTK** (64 k)
RAM interne Mém 1*	10 Ko	10 Ko	10 Ko
RAM externe Mém 2*	32 Ko	32 Ko	64 Ko
EEPROM interne	32 Ko	32 Ko	32 Ko***
EEPROM externe	32 Ko	32 Ko	64 Ko
Taille maximale du programme	32 Ko	32 Ko	64 Ko
Sauvegarde externe maximale	32 Ko	32 Ko	64 Ko

(\*) Mém 1 et Mém 2 en utilisation mémoire.

(\*\*) dans ce cas la cartouche 64 Ko doit être installée sur Twido et déclarée dans la configuration, si elle n'est pas déjà déclarée,

(\*\*\*) réservé à la sauvegarde des premiers 512 mots %MW ou des premiers 256 mots doubles %MD.

## Sauvegarde et restauration sans cartouche de sauvegarde, ni cartouche de mémoire étendue

### Introduction

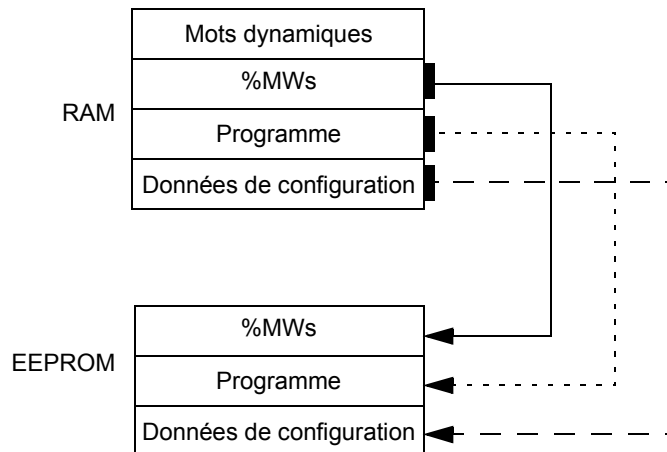
Ce sous-chapitre détaille les fonctions de sauvegarde et de restauration de la mémoire dans les automates modulaires et compacts sans cartouche de mémoire, ni cartouche de mémoire étendue branchée.

### Présentation

Les programmes Twido, les mots mémoire et les données de configuration peuvent être sauvegardés à l'aide de l'EEPROM interne des automates. Etant donné que la sauvegarde d'un programme dans l'EEPROM interne efface tout mot mémoire préalablement sauvegardé, sauvegardez tout d'abord le programme, puis les mots mémoire configurés. Les données dynamiques peuvent être stockées dans les mots mémoire, puis sauvegardées dans l'EEPROM. Si aucun programme n'est sauvegardé dans l'EEPROM interne, vous ne pouvez pas y sauvegarder des mots mémoire.

### Structure de la mémoire

Ci-dessous est présenté un schéma de la structure de mémoire d'un automate. Les flèches montrent les éléments pouvant être sauvegardés dans l'EEPROM depuis la RAM :



**Sauvegarde du programme**

Pour sauvegarder votre programme dans l'EEPROM, procédez comme suit.

Etape	Action
1	L'élément suivant doit être vrai : Le programme dans la RAM est valide.
2	Dans la fenêtre du logiciel Twido, déroulez le menu Automate et cliquez sur Sauvegarder.

**Restauration du programme**

Lors de la mise sous tension, il existe une méthode pour restaurer le programme dans la RAM depuis l'EEPROM (si aucune cartouche ou mémoire étendue n'est en place) :

- Le programme de la RAM n'est pas valide

Pour restaurer un programme manuellement depuis l'EEPROM, procédez comme suit :

- Dans la fenêtre du logiciel Twido, déroulez le menu Automate et cliquez sur Restituer.

**Sauvegarde des données (%MWs)**

Pour sauvegarder vos données (mots mémoire) dans l'EEPROM, procédez comme suit :

Etape	Action
1	Les éléments suivants doivent être vrais : Un programme valide est présent dans la RAM (%SW96:X6=1). Le même programme valide est déjà sauvegardé dans l'EEPROM. Les mots mémoire sont configurés dans le programme.
2	Définir %SW97 par rapport à la longueur des mots mémoire à sauvegarder. <b>Remarque</b> : La longueur ne peut pas dépasser la longueur du mot mémoire configuré et doit être supérieure à 0, mais inférieure ou égale à 512.
3	Définir %SW96:X0 sur 1.

**Restauration des données (%MWs)**

Restaurer %MWs manuellement en définissant le bit système %S95 sur 1.

Les éléments suivants doivent être vrais :

- Une application de sauvegarde valide est présente dans l'EEPROM.
- L'application dans la RAM correspond à l'application de sauvegarde dans l'EEPROM.
- Les mots mémoire de sauvegarde sont valides.

## Sauvegarde et restauration avec une cartouche de sauvegarde de 32 K

---

### Introduction

Ce sous-chapitre détaille les fonctions de sauvegarde et de restauration de la mémoire des automates modulaires et compacts équipés d'une cartouche de sauvegarde de 32 K.

---

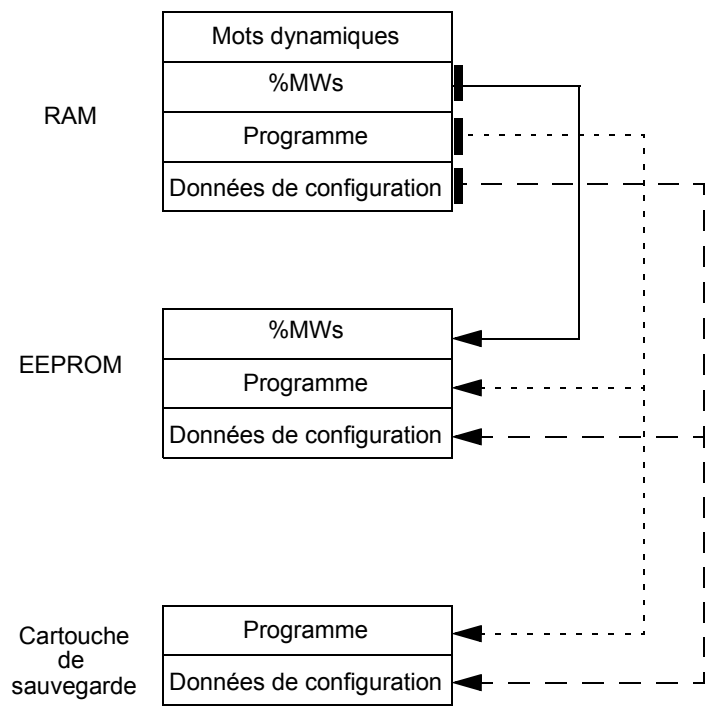
### Présentation

La cartouche de sauvegarde est utilisée pour sauvegarder un programme et le transférer vers d'autres automates Twido. Elle doit être retirée d'un automate et mise de côté une fois le programme installé ou sauvegardé. Seules les données du programme et les données de configuration peuvent être sauvegardées dans la cartouche (%MWs ne peut pas être sauvegardé dans la cartouche de sauvegarde de 32 K). Les données dynamiques peuvent être stockées dans les mots mémoire, puis sauvegardées dans l'EEPROM. Une fois l'installation du programme terminée, tout %MWs sauvegardé dans l'EEPROM interne avant l'installation sera perdu.

---

Structure de la mémoire

Ci-dessous est présenté un schéma de la structure de mémoire d'un automate avec une cartouche de sauvegarde branchée. Les flèches montrent les éléments pouvant être sauvegardés dans l'EEPROM et la cartouche depuis la RAM :



Sauvegarde du programme

Pour sauvegarder votre programme dans la cartouche de sauvegarde, procédez comme suit :

Etape	Action
1	Mettez l'automate hors tension.
2	Raccordez la cartouche de sauvegarde.
3	Mettez l'automate sous tension.
4	Dans la fenêtre du logiciel Twido, déroulez le menu Automate et cliquez sur Sauvegarder.
5	Mettez l'automate hors tension.
6	Retirez la cartouche de sauvegarde de l'automate.

## Restauration du programme

Pour charger un programme sauvegardé sur une cartouche de sauvegarde dans un automate, procédez comme suit :

Etape	Action
1	Mettez l'automate hors tension.
2	Raccordez la cartouche de sauvegarde.
3	Mettez l'automate sous tension. (Si le démarrage automatique est configuré, vous devez à nouveau effectuer la mise sous tension pour entrer en mode d'exécution.)
4	Mettez l'automate hors tension.
5	Retirez la cartouche de sauvegarde de l'automate.

## Sauvegarde des données (%MWs)

Pour sauvegarder vos données (mots mémoire) dans l'EEPROM, procédez comme suit :

Etape	Action
1	Les éléments suivants doivent être vrais : Un programme valide est présent dans la RAM. Le même programme valide est déjà sauvegardé dans l'EEPROM. Les mots mémoire sont configurés dans le programme.
2	Définir %SW97 par rapport à la longueur des mots mémoire à sauvegarder. <b>Remarque</b> La longueur ne peut pas dépasser la longueur du mot mémoire configuré et doit être supérieure à 0, mais inférieure ou égale à 512.
3	Définir %SW96:X0 sur 1.

## Restauration des données (%MWs)

Restaurez %MWs manuellement en définissant le bit système %S95 sur 1.

Les éléments suivants doivent être vrais :

- Une application de sauvegarde valide est présente dans l'EEPROM.
- L'application dans la RAM correspond à l'application de sauvegarde dans l'EEPROM.
- Les mots mémoire de sauvegarde sont valides.

## Utilisation de la cartouche de mémoire étendue de 64 K

---

### Introduction

Ce sous-chapitre détaille l'utilisation des fonctions de mémoire dans les automates modulaires équipés d'une cartouche de mémoire étendue de 64 K.

---

### Présentation

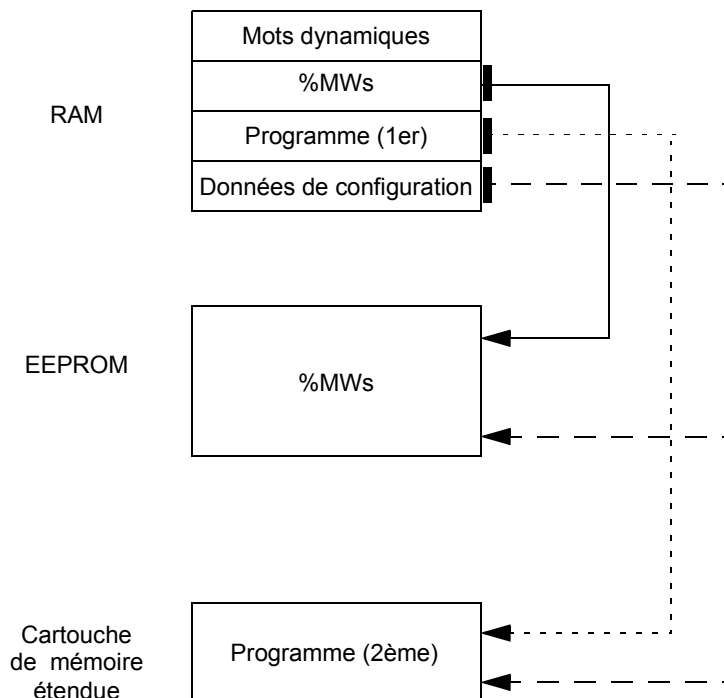
La cartouche de mémoire étendue de 64 K est utilisée pour étendre la capacité de mémoire du programme de votre automate Twido de 32 à 64 K. Elle doit rester raccordée à l'automate tant que le programme étendu est utilisé. Si la cartouche est retirée, l'automate s'arrête. Les mots mémoire sont encore sauvegardés dans l'EEPROM de l'automate. Les données dynamiques peuvent être stockées dans les mots mémoire, puis sauvegardées dans l'EEPROM. La cartouche mémoire étendue de 64 K présente le même comportement à la mise sous tension que la cartouche de sauvegarde 32 K.

---



**Structure de la mémoire**

Ci-dessous est présenté un schéma de la structure de mémoire d'un automate utilisant une cartouche mémoire étendue. Les flèches indiquent les éléments sauvegardés dans l'EEPROM et la cartouche de mémoire étendue de 64 K depuis la RAM :

**Configuration du logiciel et installation de la mémoire étendue**

Avant de procéder à l'écriture de votre programme étendu, vous devez installer la cartouche mémoire étendue de 64 K dans votre automate. Voici les quatre étapes à suivre :

Etape	Action
1	Sous l'option Matériel de la fenêtre de votre logiciel Twido, saisissez "TWDXCPMF64".
2	Mettez l'automate hors tension.
3	Raccordez la cartouche mémoire étendue de 64 K.
4	Mettez l'automate sous tension.

**Enregistrez votre programme.**

Une fois votre cartouche mémoire étendue de 64 K installée et votre programme écrit :

- Dans la fenêtre du logiciel Twido, déroulez le menu Automate et cliquez sur Sauvegarder.
- 

**Sauvegarde des données (%MWs)**

Pour sauvegarder vos données (mots mémoire) dans l'EEPROM, procédez comme suit :

Etape	Action
1	Les éléments suivants doivent être vrais : Un programme valide est présent. Les mots mémoire sont configurés dans le programme.
2	Définir %SW97 par rapport à la longueur des mots mémoire à sauvegarder. <b>Remarque</b> : La longueur ne peut pas dépasser la longueur du mot mémoire configuré et doit être supérieure à 0, mais inférieure ou égale à 512.
3	Définir %SW96:X0 sur 1.

---

**Restauration des données (%MWs)**

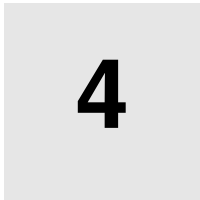
Restaurez %MWs manuellement en définissant le bit système %S95 sur 1.

Les éléments suivants doivent être vrais :

- Un programme valide est présent.
  - Les mots mémoire de sauvegarde sont valides
-

---

# Modes de fonctionnement de l'automate



---

## Présentation

**Objet de ce chapitre** Ce chapitre offre des informations sur les modes de fonctionnement des automates, ainsi que sur l'exécution cyclique et périodique de programmes. Vous y trouverez également des informations détaillées sur les coupures secteur et les opérations de restauration.

**Contenu de ce chapitre** Ce chapitre contient les sujets suivants :

Sujet	Page
Scrutation cyclique	68
Scrutation périodique	70
Vérification de la durée de scrutation	73
Modes de fonctionnement	75
Gestion des coupures et des reprises secteur	77
Gestion d'une reprise à chaud	80
Gestion d'un démarrage à froid	82
Initialisation des objets	84

## Scrutation cyclique

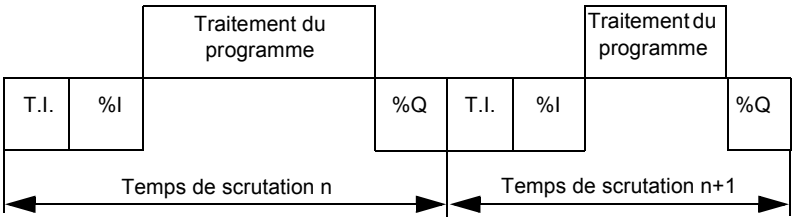
### Introduction

La scrutation cyclique consiste à enchaîner les uns après les autres les cycles automates. Après avoir effectué la mise à jour des sorties (troisième phase du cycle de tâche), le système exécute un certain nombre de ses propres tâches et déclenche immédiatement un autre cycle de tâche.

**Note :** La durée de scrutation du programme utilisateur est contrôlée par le temporisateur chien de garde de l'automate et ne doit pas dépasser 500 ms. Sinon un défaut apparaît faisant passer immédiatement l'automate en mode Halt. Sous ce mode, les sorties sont forcées sur leur état de repli par défaut.

### Fonctionnement

L'illustration suivante montre les phases d'exécution de la scrutation cyclique.



### Description des phases d'un cycle

Le tableau suivant décrit les phases d'un cycle.

Repère	Phase	Description
T.I.	traitement interne	Le système réalise implicitement la surveillance de l'automate (gestion des bits et mots système, mise à jour des valeurs courantes de l'horodateur, mise à jour des voyants d'état, détection des commutateurs RUN/STOP, etc.) et le traitement des requêtes en provenance de TwidoSoft (modifications et animation).
%I, %IW	Acquisition des entrées	Ecriture en mémoire de l'état des entrées associés aux modules TOR et analogique.
-	Traitement du programme	Exécution du programme d'application écrit par l'utilisateur.
%Q, %QW	Mise à jour des sorties	Ecriture des bits ou des mots de sorties associés aux modules TOR et analogique.

**Mode de fonctionnement**

**Automate en mode RUN**, le processeur effectue les opérations suivantes :

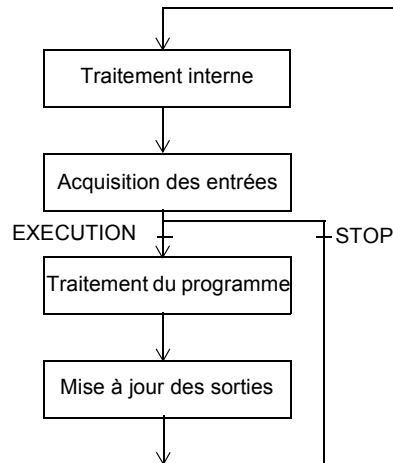
- Traitement interne
- Acquisition des entrées
- Traitement du programme d'application
- Mise à jour des sorties

**Automate en mode STOP**, le processeur effectue les opérations suivantes :

- Traitement interne
- Acquisition des entrées

**Illustration**

L'illustration suivante présente les cycles de fonctionnement.

**Contrôle du cycle**

Le contrôle du cycle est effectué par le chien de garde.

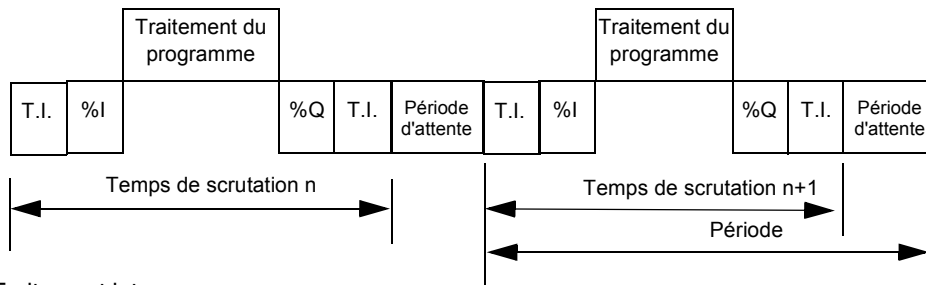
## Scrutation périodique

## Introduction

Dans ce mode de fonctionnement, l'acquisition des entrées, le traitement du programme d'application et la mise à jour des sorties s'effectuent de façon périodique selon un intervalle défini lors de la configuration (de 2 à 150 ms). Au début de la scrutation de l'automate, un temporisateur, dont la valeur est initialisée sur la période définie lors de la configuration, démarre le décomptage. La scrutation de l'automate doit se terminer avant la fin du décomptage et avant le début d'une nouvelle scrutation.

## Fonctionnement

L'illustration suivante présente les phases d'exécution de la scrutation périodique.



T.I : Traitement interne

### Description des phases de fonctionnement

Le tableau suivant décrit les phases de fonctionnement.

Repère	Phase	Description
T.I.	traitement interne	Le système réalise implicitement la surveillance de l'automate (gestion des bits et mots système, mise à jour des valeurs courantes de l'horodateur, mise à jour des voyants d'état, détection des commutateurs RUN/STOP, etc.) et le traitement des requêtes en provenance de TwidoSoft (modifications et animation).
%I, %IW	Acquisition des entrées	Ecriture en mémoire de l'état des entrées associés aux modules TOR et analogique.
-	Traitement du programme	Exécution du programme d'application écrit par l'utilisateur.
%Q, %QW	Mise à jour des sorties	Ecriture des bits ou des mots de sorties associés aux modules TOR et analogique.

**Mode de  
fonctionnement**

**Automate en mode RUN**, le processeur effectue les opérations suivantes :

- Traitement interne
- Acquisition des entrées
- Traitement du programme d'application
- Mise à jour des sorties

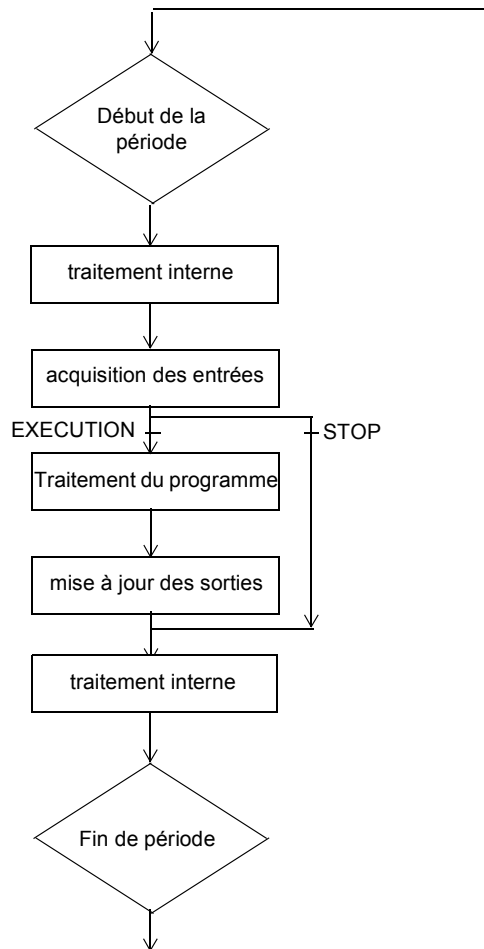
Si la période n'est pas terminée, le processeur poursuit son cycle de fonctionnement jusqu'à la fin de la période du traitement interne. Si la durée de fonctionnement dépasse celle affectée à la période, l'automate signale un débordement de période en mettant le bit système %S19 à 1. Le traitement se poursuit jusqu'à son exécution totale. Néanmoins, il ne doit pas dépasser le temps limite du chien de garde. La scrutation suivante est enchaînée après l'écriture implicite des sorties de la scrutation en cours.

**Automate en mode STOP**, le processeur effectue les opérations suivantes :

- Traitement interne
  - Acquisition des entrées
-

## Illustration

L'illustration suivante présente les cycles de fonctionnement.



## Contrôle du cycle

Deux contrôles sont effectués :

- débordement de période
- chien de garde



## Vérification de la durée de scrutation

### Généralités

Le cycle de tâche est régulé par un temporisateur chien de garde appelé Tmax (durée maximale du cycle de la tâche). Ce temporisateur permet d'afficher les erreurs de l'application (boucles infinies, etc.) et garantit une durée maximale du rafraîchissement des sorties.

### Chien de garde logiciel (fonctionnement périodique ou cyclique)

Au cours du fonctionnement périodique ou cyclique, le déclenchement du chien de garde provoque une erreur logicielle. L'application passe en mode HALT et le bit système %S11 est mis à 1. La relance de la tâche nécessite une connexion à TwidoSoft afin d'analyser la cause de l'erreur, une modification de l'application pour corriger l'erreur, puis une remise en RUN du programme.

**Note :** L'état HALT correspond à l'arrêt immédiat de l'application causé par une erreur d'application logicielle, telle qu'un débordement de scrutation. Les données gardent les valeurs courantes, permettant ainsi l'analyse de la cause de l'erreur. Le programme s'arrête sur l'instruction en cours. La communication avec l'automate est disponible.

### Contrôle en fonctionnement périodique

En fonctionnement périodique, un contrôle supplémentaire permet de détecter un dépassement de période :

- **%S19** indique que la période est dépassée. Il est mis à :
  - 1 par le système lorsque la durée de scrutation est supérieure à la durée de la tâche,
  - 0 par l'utilisateur.
- **%SW0** contient la valeur de la période (0-150 ms). Il est :
  - initialisé lors d'un démarrage à froid par la valeur choisit au moment de la configuration et,
  - peut être modifié par l'utilisateur.

**Exploitation des  
temps  
d'exécution de la  
tâche maître**

Les mots système suivants permettent d'obtenir des informations sur le temps de cycle de l'automate :

- **%SW11** initialise la durée maximale du chien de garde (10 à 500 ms).
- **%SW30** contient le durée d'exécution du dernier cycle de scrutation de l'automate.
- **%SW31** contient la durée d'exécution du plus long cycle de scrutation de l'automate depuis le dernier démarrage à froid.
- **%SW32** contient la durée d'exécution du plus court cycle de scrutation de l'automate depuis le dernier démarrage à froid.

<p><b>Note :</b> Ces différentes informations sont également accessibles depuis l'éditeur de configuration.</p>
---

## Modes de fonctionnement

---

### Introduction

Twido Soft est utilisé pour prendre en compte les trois groupes de modes de fonctionnement :

- vérification
- exécution ou production
- arrêt.

### Démarrage via Grafcet

Ces différents modes de fonctionnement sont accessibles depuis Grafcet ou en utilisant Grafcet, en appliquant les méthodes suivantes :

- initialisation de Grafcet
- préréglage des étapes
- conservation d'une situation
- gel de diagrammes.

Le traitement préliminaire et l'utilisation de bits système garantissent une gestion efficace du mode de fonctionnement qui ne provoque aucune complication du programme utilisateur et qui n'implique aucune surcharge sur ce dernier.

---

## Bits système Grafcet

L'utilisation des bits %S21, %S22 et %S23 est réservée au traitement préliminaire. Ces bits sont automatiquement remis à zéro par le système, et ne doivent être écrits que par l'instruction Set **S**.

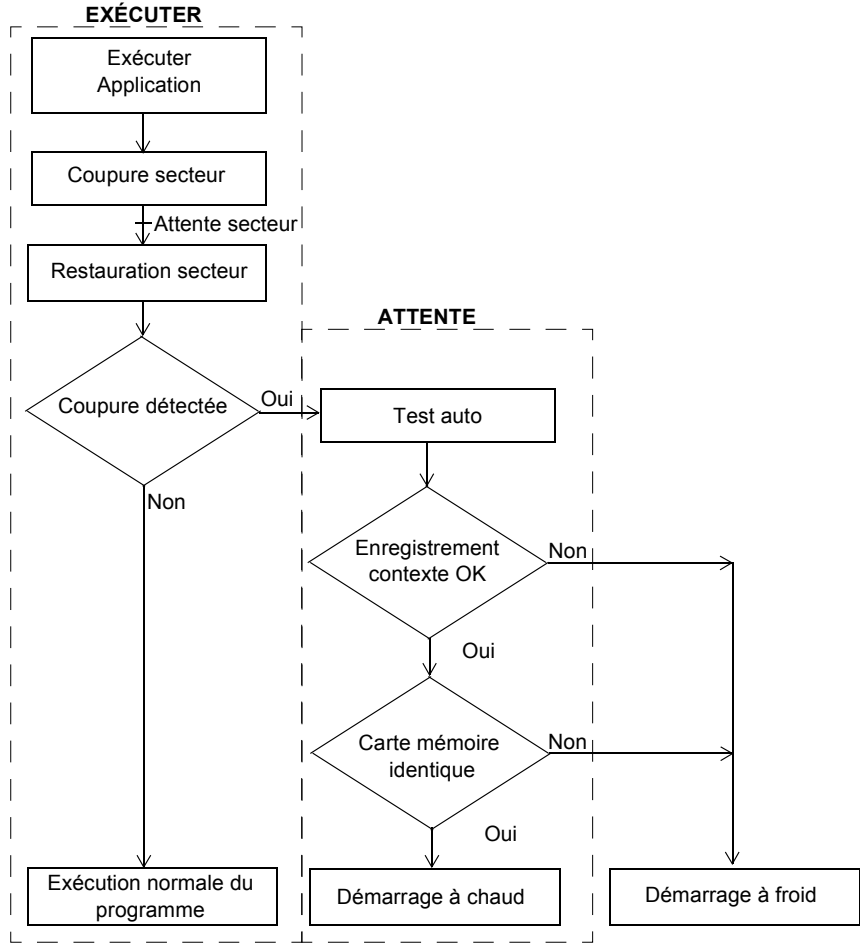
Le tableau suivant présente les bits système associés à Grafcet :

Bit	Fonction	Description
%S21	Initialisation du GRAFCET	<p>Normalement à 0, ce bit est mis à 1 par :</p> <ul style="list-style-type: none"> <li>• un démarrage à froid, %S0=1 ;</li> <li>• l'utilisateur, uniquement dans la section du programme de prétraitement, à l'aide de l'instruction Set <b>S</b> %S21 ou d'une bobine Set <b>-(S)-</b> %S21.</li> </ul> <p>Conséquences:</p> <ul style="list-style-type: none"> <li>• Désactivation de toutes les étapes.</li> <li>• Activation de toutes les étapes initiales.</li> </ul>
%S22	GRAFCET RESET	<p>Normalement mis à 0, ce bit peut être mis à 1, uniquement par le programme au cours du prétraitement.</p> <p>Conséquences :</p> <ul style="list-style-type: none"> <li>• Désactivation de toutes les étapes.</li> <li>• Arrêt de la scrutation du traitement séquentiel.</li> </ul>
%S23	Prépositionnement et gel du GRAFCET	<p>Normalement mis à 0, ce bit peut être mis à 1, uniquement par le programme au cours du prétraitement.</p> <ul style="list-style-type: none"> <li>• Prépositionnement en mettant %S22 à 1.</li> <li>• Prépositionne les étapes pour leur activation, par une série d'instructions S Xi.</li> <li>• Activation du prépositionnement en mettant %S23 à 1.</li> </ul> <p>Gel d'une situation :</p> <ul style="list-style-type: none"> <li>• Dans la situation initiale : par le maintien de %S21 à 1 par le programme.</li> <li>• Dans une situation « vide » : par le maintien de %S22 à 1 par le programme.</li> <li>• Dans une situation déterminée par le maintien de %S23 à 1.</li> </ul>

## Gestion des coupures et des reprises secteur

### Illustration

L'illustration suivante présente les différentes reprises secteur détectées par le système. Si la durée de la coupure est inférieure au temps de filtrage de l'alimentation (environ 10 ms pour une alimentation en courant alternatif ou 1 ms pour une alimentation en courant continu), elle n'est pas prise en compte par le programme qui s'exécute normalement.



**Note :** Le contexte est enregistré dans une mémoire RAM sur batterie de secours. A la mise sous tension, le système vérifie l'état des batteries et du contexte enregistré afin de déterminer si un démarrage à chaud est possible.

### Bit d'entrée Run/Stop et option Démarrage automatique en Run

Le bit d'entrée Run/Stop est prioritaire sur l'option "Démarrage automatique en Run" accessible à partir de la boîte de dialogue Mode de scrutation. Si le bit Run/Stop est à 1, l'automate redémarre en mode Run à la reprise secteur. Le mode de l'automate est déterminé de la façon suivante.

Bit d'entrée Run/Stop	Démarrage automatique en Run	Etat résultant
Zéro	Zéro	Stop
Zéro	Un	Stop
Front montant	Sans importance	Run
Un	Sans importance	Run
Non configuré dans le logiciel	Zéro	Stop
Non configuré dans le logiciel	Un	Run

**Note :** Pour tous les automates compacts avec une version logicielle V1.0, si l'automate est en mode Run à l'interruption du secteur et que l'indicateur "Démarrage automatique en Run" n'est pas sélectionné dans la boîte de dialogue Mode de scrutation, l'automate redémarre en mode Stop à la reprise secteur, dans le cas contraire il redémarre à froid.

**Note :** Pour tous les automates modulaires et compacts avec une version logicielle V1.11, si la batterie de l'automate fonctionne normalement lors de l'interruption du secteur, l'automate redémarre dans le mode effectif au moment de l'interruption. L'indicateur "Démarrage automatique en Run", sélectionné dans la boîte de dialogue Mode de scrutation, n'aura aucun effet sur le mode adopté à la reprise secteur.

**Fonctionnement**    Le tableau suivant décrit les phases du traitement des coupures secteur.

Phase	Description
1	Lors de la coupure secteur, le système mémorise le contexte application et l'heure de la coupure.
2	Il met toutes les sorties dans l'état de repli (état 0).
3	<p>A la reprise secteur, le contexte sauvegardé est comparé à celui en cours. Cette comparaison permet de définir le type de démarrage à exécuter :</p> <ul style="list-style-type: none"> <li>● Si le contexte application a changé (perte du contexte système ou nouvelle application), l'automate procède à l'initialisation de l'application : démarrage à froid (systématique pour le compact).</li> <li>● Si le contexte application est identique, l'automate effectue une reprise sans initialisation des données : redémarrage à chaud.</li> </ul>

## Gestion d'une reprise à chaud

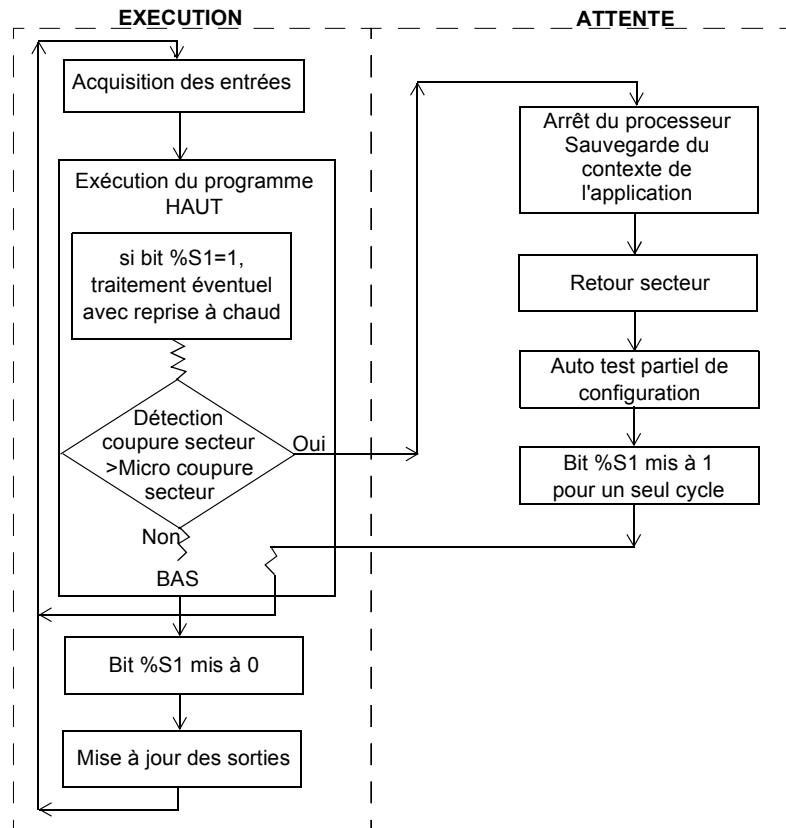
### Cause d'une reprise à chaud

Une reprise à chaud peut être provoquée :

- par une reprise secteur sans perte du contexte,
- lorsque le bit système **%S1** est mis à 1 par le programme,
- depuis l'afficheur, lorsque l'automate est en mode STOP.

### Illustration

Le schéma ci-après décrit le fonctionnement d'une reprise à chaud en mode d'exécution (RUN).





**Reprise de l'exécution du programme**

Le tableau suivant décrit les phases de reprise de l'exécution d'un programme après une reprise à chaud.

Phase	Description
1	L'exécution du programme reprend à partir de l'élément où a eu lieu la coupure secteur, sans mise à jour des sorties. <b>Remarque :</b> Seuls les éléments du code de l'utilisateur sont redémarrés. Le code système (la mise à jour des sorties, par exemple) n'est pas redémarré.
2	A la fin du cycle de reprise, le système : <ul style="list-style-type: none"> <li>● annule la réservation de l'application lorsqu'elle est réservée (et provoque une application STOP en cas de débogage) ;</li> <li>● effectue la réinitialisation des messages.</li> </ul>
3	Le système effectue un cycle de reprise au cours duquel il : <ul style="list-style-type: none"> <li>● relance la tâche avec les bits <b>%S1</b> (indicateur de reprise à chaud) et <b>%S13</b> (premier cycle en mode RUN) mis à 1,</li> <li>● remet à l'état 0 les bits <b>%S1</b> et <b>%S13</b> à la fin de ce premier cycle de la tâche.</li> </ul>

**Gestion d'un démarrage à chaud**

En cas de démarrage à chaud et lorsque le traitement d'une application particulière est requis, le bit **%S1** doit être testé en début du cycle de tâche et le programme correspondant doit être appelé.

**Sorties après une coupure secteur**

Dès qu'une coupure secteur est détectée, les sorties sont mis dans un état de repli (par défaut) de 0.  
A la reprise secteur, les sorties conservent leur dernier état jusqu'à ce qu'elles soient remises à jour par la tâche.

## Gestion d'un démarrage à froid

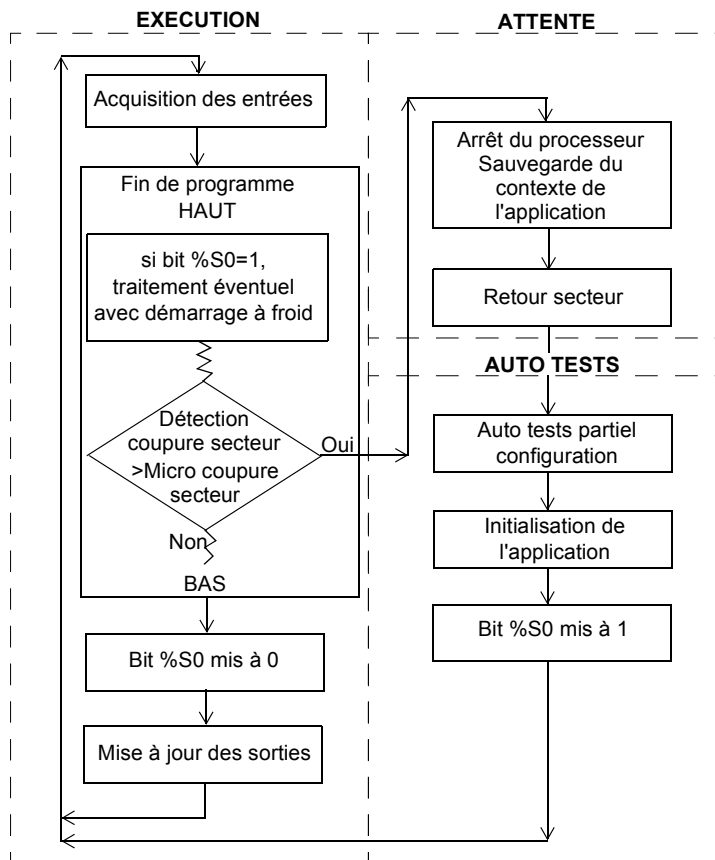
### Cause d'un démarrage à froid

Un démarrage à froid peut être provoqué :

- par le chargement d'une nouvelle application dans la mémoire RAM,
- par une reprise secteur avec perte du contexte de l'application,
- lorsque le bit **%S0** est mis à 1 par le programme,
- depuis l'afficheur, lorsque l'automate est en mode STOP.

### Illustration

Le dessin suivant décrit le fonctionnement d'une reprise à froid en mode d'exécution (RUN).



**Fonctionnement**

Le tableau ci-après décrit les phases de reprise de l'exécution du programme sur reprise à froid.

Phase	Description
1	A la mise sous tension, l'automate est en mode d'exécution (RUN). En cas de redémarrage faisant suite à un arrêt causé par une erreur, le système impose un redémarrage à froid. L'exécution du programme reprend en début de cycle.
2	Le système effectue : <ul style="list-style-type: none"> <li>• une remise à 0 des bits et des mots internes et des images E/S,</li> <li>• l'initialisation des bits et mots système,</li> <li>• l'initialisation des blocs fonction à partir des données de configuration.</li> </ul>
3	Pour ce premier cycle de reprise, le système : <ul style="list-style-type: none"> <li>• relance la tâche avec les bits <b>%S0</b> (indicateur de reprise à froid) et <b>%S13</b> (premier cycle en mode RUN) mis à 1,</li> <li>• remet à 0 les bits <b>%S0</b> et <b>%S13</b> à la fin de ce premier cycle de tâche.</li> <li>• remet à 0 les bits <b>%S31</b>, <b>%S38</b> et <b>%S39</b> (indicateurs de contrôle d'événement), et le mot <b>%SW48</b> (nombre d'événements exécutés).</li> </ul>

**Gestion d'un démarrage à froid**

Dans le cas d'un démarrage à froid et lorsque le traitement particulier d'une application est requis, le bit **%S0** (qui est à 1) doit être testé au cours du premier cycle de la tâche.

**Sorties après une coupure secteur**

Dès qu'une coupure secteur est détectée, les sorties sont réglées sur un état de repli (par défaut) de 0.  
A la reprise secteur, les sorties sont à zéro jusqu'à ce qu'elles soient remises à jour par la tâche.

## Initialisation des objets

---

### Introduction

Les automates peuvent être initialisés par TwidoSoft en mettant à 1 les bits système **%S0** (démarrage à froid) et **%S1** (reprise à chaud).

---

### Initialisation en démarrage à froid

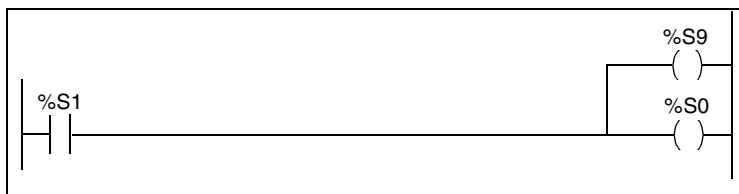
Pour une initialisation en démarrage à froid, le bit système **%S0** doit être mis à 1.

---

### Initialisation des objets (identique que démarrage à froid) à la mise sous tension à l'aide de %S0 et de %S1

Pour une initialisation des objets à la mise sous tension, les bits système **%S1** et **%S0** doivent être mis à 1.

L'exemple suivant montre comment programmer une initialisation des objets lors d'une reprise à chaud à l'aide des bits système.



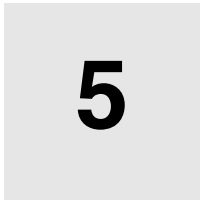
- LD %S1 Si %S1 = 1 (reprise à chaud), le bit %S0 est mis à 1 ce qui initialise l'automate.  
 ST %S0 Ces deux bits sont remis à zéro par le système à la fin de la scrutation suivante.  
 ST %S9 Ce bit est utilisé pour initialiser les sorties.

**Note :** Ne mettez pas %S0 à 1 pour plus d'une scrutation de l'automate.

---

---

# Gestion des tâches événementielles



---

## En bref...

**Présentation** Ce chapitre décrit les tâches événementielles et leur exécution dans l'automate.

**Note :** Les tâches événementielles ne sont pas gérées par l'automate Twido Brick 10 (TWDLCAA10DRF).

**Contenu de ce chapitre** Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation des tâches événementielles	86
Description des différentes sources d'événement	87
Gestion des événements	89

## Présentation des tâches événementielles

---

### Introduction

Le précédent chapitre présente les tâches périodiques (Voir *Scrutation périodique*, p. 70) et cycliques (Voir *Scrutation cyclique*, p. 68) où les objets sont mis à jour en début et fin de tâche. Des sources d'événements peuvent provoquer des interruptions de cette tâche pendant lesquelles des tâches plus prioritaires (événementielles) sont exécutées pour permettre une mise à jour plus rapide des objets.

Une tâche événementielle :

- est une portion de programme exécutée à une condition donnée (source d'événement),
  - possède une priorité plus haute que le programme principal,
  - garantit un temps de réponse rapide qui permet de réduire le temps de réponse du système.
- 

### Description d'un événement

Un événement se compose :

- d'une source d'événement qui peut être défini comme une condition d'interruption logicielle ou matérielle qui interrompt le programme principal (Voir *Description des différentes sources d'événement*, p. 87),
  - d'une section qui est une entité autonome de programmation liée à un événement,
  - d'une file d'événements permettant de stocker la liste des événements jusqu'à leur exécution,
  - d'une priorité qui est l'ordre d'exécution de l'événement.
-

## Description des différentes sources d'événement

### Présentation des différentes sources d'événement

Une source d'événement nécessite d'être gérée par le logiciel pour assurer l'interruption du programme principal par l'événement et l'appel de la section de programmation liée à l'événement. Le temps de scrutation de l'application n'a pas d'effet sur l'exécution des événements.

Les 9 sources d'événements permises sont les suivantes :

- 4 conditions liées aux seuils des blocs fonction compteur rapide (2 événements par instance de %VFC),
- 4 conditions liées aux entrées physiques d'une base automate,
- 1 condition périodique.

Une source d'événement ne peut être attachée qu'à un seul événement et doit être immédiatement détectée par le logiciel TwidoSoft. Sitôt détectée, le logiciel exécute la section de programmation attachée à l'événement : chaque événement est attaché à un sous-programme portant une étiquette **SRi**: définie lors de la configuration des sources d'événement.

### Événement sur entrées physiques d'une base automate

Les entrées %I0.2, %I0.3, %I0.4 et %I0.5 peuvent être utilisées comme sources d'événement, à condition qu'elles ne soient pas verrouillées et que les événements y soient permis pendant la configuration.

Les traitements événementiels peuvent être déclenchés par les entrées 2 à 5 d'une base automate (position 0), sur front montant ou descendant.

Pour plus de détails sur la configuration de l'événement, consultez la section "Configuration matérielle -> Configuration des entrées" dans l'aide en ligne du "Guide d'exploitation TwidoSoft".

### Événement sur les sorties d'un bloc fonction %VFC

Les sorties TH0 et TH1 du bloc fonction %VFC sont des sources d'événements. Les sorties TH0 et TH1 passent respectivement :

- à 1 quand la valeur est supérieure au seuil S0 et au seuil S1,
- à 0 quand la valeur est inférieure au seuil S0 et au seuil S1.

Un front montant ou descendant de ces sorties peut déclencher un traitement événementiel.

Pour plus de détails sur la configuration de l'événement, consultez la section "Configuration logicielle -> Compteurs rapides" dans l'aide en ligne du "Guide d'exploitation TwidoSoft".

**Événement  
périodique**

Cet événement exécute une même section de programmation de façon périodique. Cette tâche est plus prioritaire que la tâche principale (maître).

Cette source d'événement est moins prioritaire par contre que les autres sources d'événement.

La période de cette tâche est fixée en configuration, de 5 à 255 ms. Un seul événement périodique peut être utilisé.

Pour plus de détails sur la configuration de l'événement, consultez la section "Configuration des paramètres du programme -> Mode de scrutation" dans l'aide en ligne du "Guide d'exploitation TwidoSoft" .

---



## Gestion des événements

### File d'événements et priorité

Les événements présentent 2 priorités possibles : Haute ou Basse. Mais un **seul** type d'événement (donc une seule source d'événement) peut avoir la priorité Haute. Les autres événements ont alors une priorité Basse, et leur ordre d'exécution dépend alors de leur ordre de détection.

Pour gérer l'ordre d'exécution des tâches événementielles, il existe deux files d'événements :

- l'une permettant de stocker jusqu'à 16 événements de priorité Haute (d'une même source d'événement),
- l'autre permettant de stocker jusqu'à 16 événements de priorité Basse (des autres sources d'événement).

Ces files sont gérées comme des FIFO (First In First Out) : le premier événement stocké est le premier exécuté. Mais elles ne peuvent stocker que 16 événements, les événements supplémentaires sont perdus.

La file de priorité Basse n'est exécutée que lorsque la file de priorité Haute est vide.

### Gestion des files d'événements

A chaque fois qu'une interruption apparaît (liée à une source d'événement), la séquence suivante est lancée :

Etape	Description
1	Gestion de l'interruption : <ul style="list-style-type: none"> <li>• connaissance de l'interruption physique,</li> <li>• événement stocké dans la file d'événements appropriée,</li> <li>• vérification qu'un événement de même priorité n'est pas en cours (sinon l'événement reste en attente dans sa file).</li> </ul>
2	Sauvegarde du contexte.
3	Exécution de la section de programmation (sous-programme étiqueté SRI:) liée à l'événement.
4	Mise à jour des sorties
5	Restauration du contexte

Avant que le contexte ne soit rétabli, tous les événements de la file doivent être exécutés.

**Contrôle des événements**

Des bits et mots systèmes sont utilisés pour contrôler les événements (Voir *Bits système et mots système*, p. 545) :

- %S31 : permet d'exécuter ou de retarder un événement,
- %S38 : permet de placer ou non un événement dans la file d'événements,
- %S39 : permet de savoir si des événements sont perdus,
- %SW48 : contient le nombre d'événements qui ont été exécutés depuis la dernière reprise à froid.

La valeur des bits et mots est initialisée à zéro lors d'une reprise à froid ou après chargement d'une application, mais reste inchangée lors d'une reprise à chaud. Dans tous les cas la file d'événements est initialisée.

---

---

## Fonctions spéciales



---

### Présentation

#### Objet de cette partie

Cette rubrique décrit les différents types de communications, les fonctions analogiques intégrées, la gestion des modules d'E/S analogiques et la mise en œuvre du bus AS-Interface V2 des automates Twido.

#### Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
6	Communications	93
7	Fonctions analogiques intégrées	195
8	Gestion des modules analogiques	199
9	Mise en œuvre du bus AS-Interface V2	209
10	Fonctionnement de l'afficheur	245



---

# Communications

# 6

---

## Présentation

### Objet de ce chapitre

Ce chapitre offre une présentation des procédures de configuration, de programmation et de gestion des différents types de communications à l'aide d'automates Twido.

---

**Contenu de ce chapitre**

Ce chapitre contient les sujets suivants :

<b>Sujet</b>	<b>Page</b>
Présentation des différents types de communications	95
Communications entre TwidoSoft et l'automate	97
Communication entre TwidoSoft et un modem	103
Communications de liaison distante	114
Communications ASCII	127
Communications Modbus	138
Requêtes Modbus standard	153
Vue d'ensemble des communications TCP/IP Ethernet	159
Guide de configuration rapide TCP/IP pour les communications Ethernet PC vers automate	160
Connexion de l'automate au réseau	166
Adressage IP	167
Affectation d'adresses IP	169
Configuration TCP/IP	173
Onglet Configurer adresse IP	175
Onglet IP repérée	177
Onglet Contrôle d'inactivité	179
Onglet Périphériques distants	181
Affichage de la configuration Ethernet	183
Gestion des connexions Ethernet	185
Voyants Ethernet	188
Messagerie Modbus TCP	190

---

## Présentation des différents types de communications

---

### Présentation

Twido dispose d'un ou deux ports série de communication utilisés pour communiquer avec les automates E/S distants, les automates d'extension ou divers périphériques. Les deux ports, lorsqu'ils sont disponibles, peuvent être utilisés pour tous les services, à l'exception de la communication avec TwidoSoft, qui ne peut se faire qu'avec le premier port. Trois protocoles de base sont pris en charge sur chaque automate Twido : liaison distante, ASCII ou Modbus (maître ou esclave Modbus).

En outre, l'automate compact TWDLCAE40DRF dispose d'un port de communication Ethernet RJ-45. Il prend en charge le protocole client/serveur Modbus TCP/IP pour les communications poste à poste entre les automates sur le réseau Ethernet.

---

### Liaison distante

La liaison distante est un bus maître/esclave très rapide conçu pour transmettre une petite quantité de données entre l'automate maître et un maximum de sept automates distants (esclave). Les données de l'application ou les données d'E/S sont transférées en fonction de la configuration des automates distants. Il est possible d'associer différents types d'automate, tels que des automates d'E/S distantes et des automates d'extension.

---

### ASCII

Le protocole ASCII est un protocole semi-duplex en mode caractères simples utilisé pour transmettre et/ou recevoir une chaîne de caractères de/vers un périphérique (imprimante ou terminal). Ce protocole est uniquement pris en charge via l'instruction "EXCH".

---

**Modbus**

Le protocole Modbus est un protocole maître/esclave qui permet à un maître uniquement d'obtenir des réponses provenant des esclaves ou d'agir sur requête. Le maître peut s'adresser aux esclaves individuellement ou envoyer un message de diffusion générale à tous les esclaves. Les esclaves renvoient un message (réponse) aux requêtes qui leur sont adressées individuellement. Les réponses aux requêtes de diffusion générale du maître ne sont pas renvoyées.

**Maître Modbus** - Le mode maître Modbus permet à l'automate Twido d'envoyer à un esclave une requête Modbus et d'attendre sa réponse. Le mode maître Modbus est uniquement pris en charge via l'instruction "EXCH". Il gère les modes ASCII et RTU Modbus.

**Esclave Modbus** - Le mode esclave Modbus permet à l'automate Twido de répondre aux requêtes Modbus d'un maître Modbus. Ce mode de communication est utilisé par défaut lorsqu'aucun autre type de communication n'a été configuré. L'automate Twido prend en charge les données Modbus standard, les fonctions de contrôle et les extensions de service pour l'accès aux objets. Les modes ASCII et RTU Modbus sont pris en charge en mode esclave Modbus.

**Note :** 32 périphériques (sans répéteurs) peuvent être installés sur un réseau RS-485 (1 maître et jusqu'à 31 esclaves). Les repères correspondants peuvent être compris entre 1 et 247.

---

**Modbus TCP/IP**

**Note :** Le protocole Modbus TCP/IP est uniquement pris en charge par les automates compacts TWDLCAE40DRF disposant d'une interface réseau Ethernet intégrée.

Les informations suivantes décrivent le protocole d'application Modbus (MBAP - Modbus Application Protocol).

Le protocole d'application Modbus est un protocole à sept couches permettant une communication poste à poste entre des automates programmables industriels (API) et d'autres nœuds sur un réseau LAN.

La mise en œuvre actuelle de l'automate Twido TWDLCAE40DRF utilise le protocole d'application Modbus via TCP/IP sur le réseau Ethernet. Les transactions du protocole Modbus sont des messages de type requête-réponse. Un automate peut être à la fois client et serveur selon qu'il envoie des requêtes ou qu'il reçoit des réponses.

---



## Communications entre TwidoSoft et l'automate

### Présentation


Chaque automate Twido comporte, sur son port 1, une prise terminal EIA RS-485 intégrée. Cette prise possède sa propre alimentation interne. Le port 1 doit être utilisé pour la communication avec le logiciel de programmation TwidoSoft. Aucune cartouche ou aucun module de communication en option ne peut utiliser ce port. Ce dernier est néanmoins utilisable par un modem.

Vous pouvez connecter le PC au port 1 RS-485 de l'automate Twido de plusieurs façons :

- par câble TSXPCX ;
- par ligne téléphonique : connexion MODEM.

De plus, l'automate compact TWDLCAE40DRF dispose d'un port RJ-45 pour la connexion réseau Ethernet qui peut être utilisé pour la communication avec un PC prenant en charge Ethernet et exécutant le logiciel de programmation TwidoSoft. Le PC prenant en charge Ethernet peut communiquer avec le port RJ-45 de l'automate Twido TWDLCAE40DRF de deux façons :

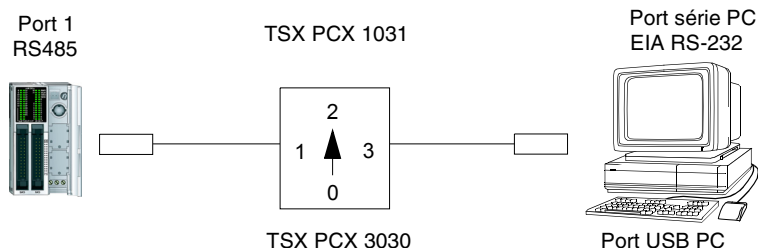
- par connexion directe via un câble inverseur UTP Ethernet RJ-45 Cat5 (déconseillé) ;
- par connexion au réseau Ethernet via un câble SFTP Ethernet RJ-45 Cat5 disponible dans le catalogue Schneider Electric (référence du câble : 490NTW000••).

	<b>ATTENTION</b>
	<p><b>RISQUE DE DETERIORATION DU MATERIEL</b></p> <p>TwidoSoft risque de ne pas détecter de déconnexion lorsque vous retirez physiquement le câble de communication TSXPCX1031, TSX PCX 3030 ou Ethernet d'un automate pour le réinsérer rapidement dans un autre automate. Afin d'éviter ce genre de problème, utilisez TwidoSoft pour effectuer la déconnexion avant de retirer le câble.</p> <p><b>Le non-respect de ces précautions peut entraîner des lésions corporelles ou des dommages matériels.</b></p>

**Raccordement  
du câble TSXPCX**

Le port EIA RS-232C ou USB de votre PC est raccordé au port 1 de l'automate à l'aide du câble de communication multifonctions TSXPCX1031 ou TSX PCX 3030. Ce câble assure la conversion des signaux entre EIA RS-232 et EIA RS-485 pour le TSX PCX 1031 et entre USB et EIA RS-485 pour le TSX PCX 3030. Il est doté d'un commutateur rotatif à quatre positions permettant de sélectionner les différents modes de fonctionnement. Les quatre positions de ce commutateur sont numérotées de 0 à 3. Pour les communications entre TwidoSoft et l'automate Twido, ce commutateur doit être positionné sur 2.

Ce raccordement est illustré dans le schéma suivant.



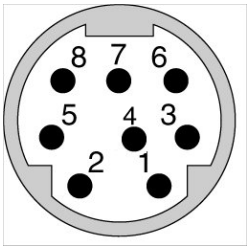
**Note :** Pour ce câble le signal DPT sur la broche 5 n'est pas mis à 0 V. Cela indique à l'automate que la connexion courante est une connexion TwidoSoft. Le signal est réglé de manière interne afin d'indiquer au microprogramme de l'automate que la connexion courante est une connexion TwidoSoft.

Brochages des connecteurs mâle et femelle

L'illustration suivante présente le brochage d'un connecteur mini DIN mâle à 8 broches et d'un bornier :

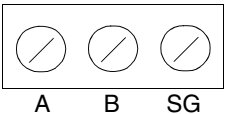
Mini DIN

TWD NAC232D, TWD NAC485D  
TWD NOZ485D, TWD NOZ232D



Bornier

TWD NAC485T  
TWD NOZ485T

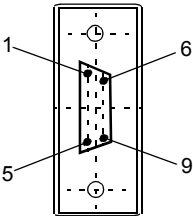


Brochages	Base RS485	Option RS485	RS232-C
1	A (+)	A (+)	RTS
2	B (-)	B (-)	DTR
3	NC	NC	TXD
4	/DE	NC	RXD
5	/DPT	NC	DSR
6	NC	NC	GND
7	0 V	0 V	GND
8	5 V	5 V	5 V

Brochages	RS485
A	A(+)
B	B(-)
SG	0V

Remarque : consommation totale maximum  
sur le 5 V (broche 8) : 180 mA

L'illustration suivante présente le brochage d'un connecteur SubD femelle à 9 broches pour le TSX PCX 1031.



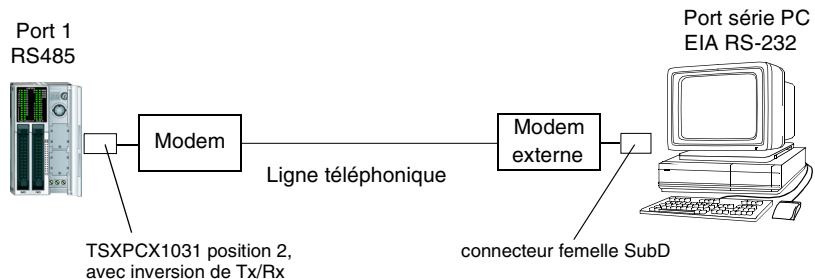
Brochages	RS232
1	DCD
2	RX
3	TX
4	DTR
5	SG
6	NC
7	RTS
8	CTS
9	NC

## Connexion par ligne téléphonique

Une connexion par modem (Voir *Communication entre TwidoSoft et un modem*, p. 103) permet de programmer et de communiquer avec un automate par ligne téléphonique.

Le modem associé à l'automate est un modem de **réception** connecté au port 1 de l'automate. Le modem associé au PC peut être interne ou externe (alors connecté au port série COM).

Ce raccordement est illustré dans le schéma suivant.



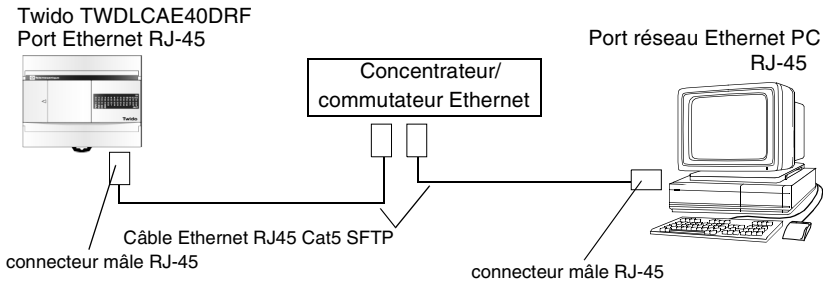
**Note :** Un seul modem peut être connecté au port 1 de l'automate.

**Note :** Attention. N'oubliez pas d'installer le logiciel fourni avec le modem, car TwidoSoft prend uniquement en compte les modems installés.

**Connexion par  
réseau Ethernet**

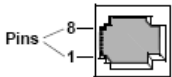
**Note :** Même si la connexion directe par câble (à l'aide d'un câble inverseur ) est prise en charge entre l'automate Twido TWDLCAE40DRF et le PC exécutant le logiciel de programmation Twidosoft, nous déconseillons cette méthode. Par conséquent, préférez toujours une connexion via un concentrateur/commutateur Ethernet.

L'illustration suivante représente une connexion entre un PC et Twido via un concentrateur/commutateur Ethernet :



**Note :** Le PC exécutant l'application TwidoSoft doit prendre en charge Ethernet.

L'automate Twido TWDLCAE40DRF dispose d'un connecteur RJ-45 pour la liaison au réseau Ethernet 100 BASE-TX prenant en charge l'autonégociation. Il prend en charge les vitesses de connexion réseau de 100 Mbit/s et 10 Mbit/s. L'illustration suivante représente le connecteur RJ-45 de l'automate Twido.



Les huit broches du connecteur RJ-45 sont positionnées verticalement et numérotées par ordre croissant du bas vers le haut. Le brochage du connecteur RJ-45 est décrit dans le tableau ci-dessous :

Brochage	Fonction	Polarité
8	NC	
7	NC	
6	RxD	(-)
5	NC	
4	NC	

Brochage	Fonction	Polarité
3	RxD	(+)
2	TxD	(-)
1	TxD	(+)

**Note :**

- Des connecteurs et brochages identiques sont utilisés pour 10Base-T et 100Base-TX.
  - Utilisez un câble Ethernet de catégorie 5 minimum pour connecter l'automate Twido à un réseau 100Base-TX.
-

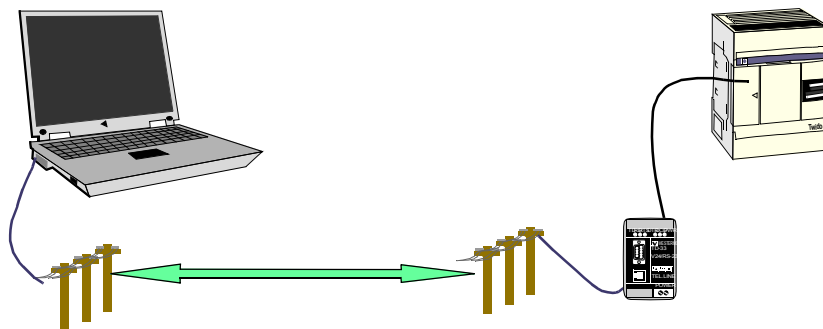
---

## Communication entre TwidoSoft et un modem

---

### Généralité

Il est possible de connecter un PC exécutant Twidosoft à un contrôleur Twido pour transférer des applications, animer des objets, exécuter des commandes mode opérateur. Il est également possible de connecter un contrôleur Twido à d'autres équipements tels qu'un autre contrôleur Twido afin d'établir une communication avec le processus d'application.



---

### Installation du modem

Tous les modems que l'utilisateur souhaite utiliser avec Twidosoft doivent être installés sous l'environnement Windows à partir de votre PC. Pour installer vos modems sous l'environnement Windows, suivez la documentation Windows. Cette installation est indépendante de Twidosoft.

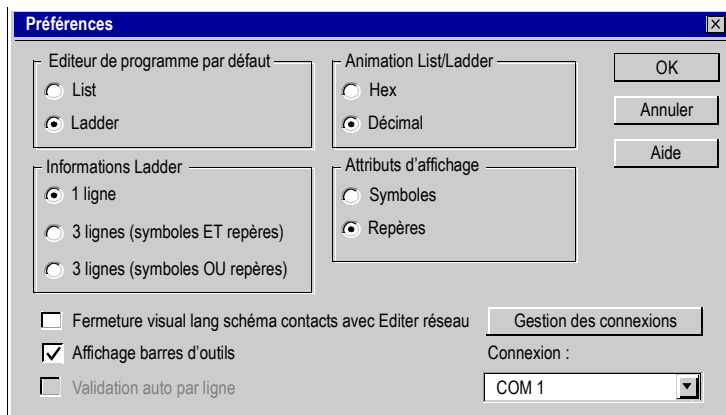
---

## Etablissement de la connexion

La connexion de communication par défaut entre Twidosoft et le contrôleur Twido est assurée par un port de communication série, utilisant le câble TSX PCX 1031 et un adaptateur croisé (voir *Annexe 1, p. 112*).

Si un modem est utilisé pour connecter le PC, alors celui-ci doit être signalé dans le logiciel Twidosoft.

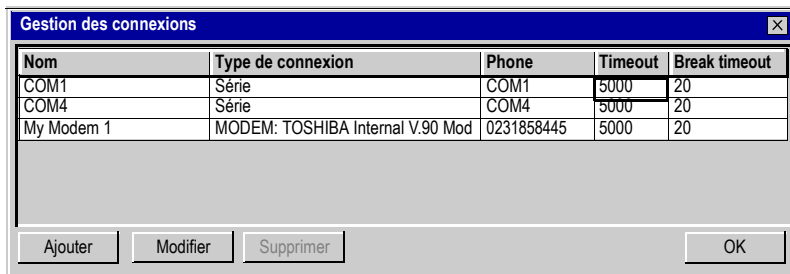
Pour sélectionner une connexion avec Twidosoft, allez dans le menu "fichier", puis "préférences".



Cet écran vous permet de sélectionner une connexion, ou de gérer des connexions, tel que création, modification, .

Pour utiliser une connexion existante, sélectionnez la parmi celles existantes dans le menu déroulant.

Si vous devez ajouter, modifier ou effacer une connexion, cliquez une fois sur "Gestion des connexions", une fenêtre s'ouvre affichant la liste des connexions et leurs propriétés.



Dans ce cas, 2 ports série sont répertoriés (Com1 et Com4) et une connexion modem utilisant un modèle TOSHIBA V.90, configuré pour composer le numéro: 0231858445 (appel national).



Vous pouvez changer le nom de chaque connexion, qui servira à la maintenance de l'application (mais le changement de COM1 ou COM4 n'est pas autorisé).

Voici le moyen de définir et sélectionner la connexion que vous voulez utiliser pour connecter votre PC à un modem.

Mais ce n'est qu'une partie des manipulations que vous devez effectuer pour établir la connexion globale entre l'ordinateur et le contrôleur Twido.

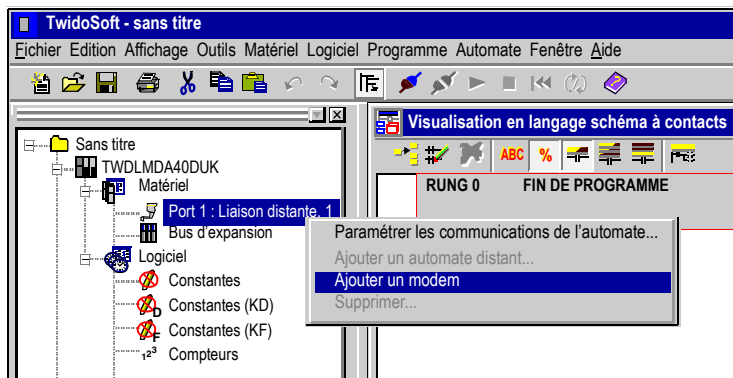
La prochaine étape concerne le contrôleur Twido. Le Twido situé à distance doit être connecté à un modem.

Tous les modems nécessitent une initialisation pour établir une connexion. Le contrôleur Twido contenant au minimum le microprogramme version V2.0 est capable d'envoyer à la mise sous tension une chaîne adaptée au modem, si le modem est configuré dans l'application.

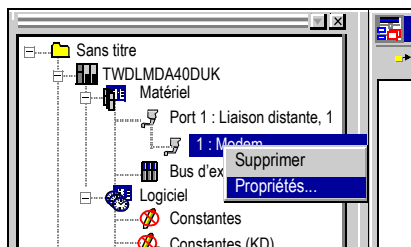
---

## Configuration du modem

Pour configurer un modem dans un contrôleur Twido, procédez ainsi:



Après avoir configuré le modem sur le port 1, nous devons définir les propriétés. Un clic droit sur le modem propose les actions "supprimer" ou "propriétés". Propriétés donne accès à une sélection d'un modem connu, création d'un nouveau, ou modification.



**Note :** La gestion du modem par le contrôleur Twido est complètement effectuée sur le port 1. Cela signifie que vous pouvez connecter un modem sur le port 2 de communication, mais dans ce cas, tous les modes opératoires et la séquence d'initialisation du modem doivent être effectués manuellement, et ne peuvent pas être effectués de la même manière que le port 1 de communication.

Prochaine étape, nous sélectionnons "propriétés" et ensuite:



Vous pouvez sélectionner un modem défini précédemment, ou en créer un nouveau en cliquant sur le bouton "...".



Donnez ensuite un nom au nouveau profil et remplissez la commandes Hayes d'initialisation comme décrit dans la documentation modem.

Sur cette image, "xxxxxx" représente la séquence d'initialisation que vous devez entrer afin de préparer le modem pour la communication adaptée, c'est à dire la vitesse en baud, la parité, le bit d'arrêt, le mode réception, .

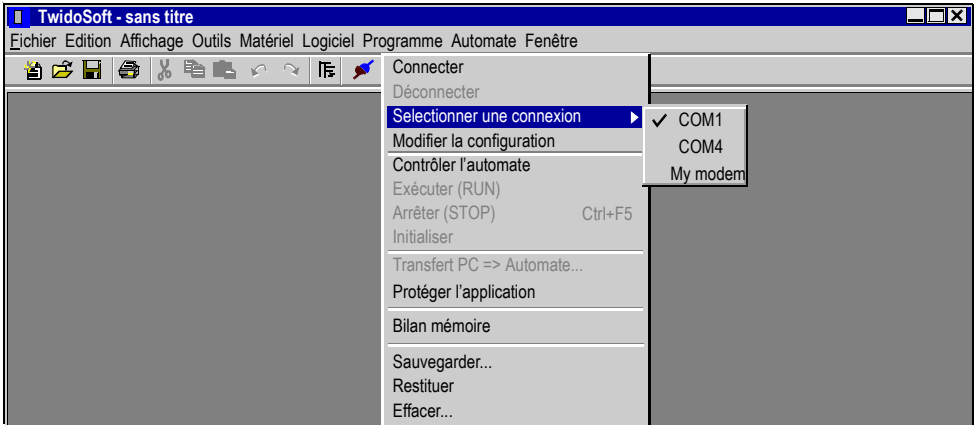
Pour compléter cette séquence vous devez vous référer à votre documentation modem.

La longueur maximum de la chaîne est : 127 caractères.

Lorsque votre application est terminée, ou au minimum lorsque le port 1 de communication est totalement décrit, transférez l'application en utilisant une "connexion point à point".

Alors le contrôleur Twido est prêt à être connecté à un PC exécutant Twidosoft par l'intermédiaire de modems.

**Séquence de connexion**                      Après avoir préparé Twidosoft et le contrôleur Twido, établissez la connexion ainsi :

Etape	Action
1	Mettez sous tension le contrôleur Twido et le modem.
2	Démarrer votre ordinateur et lancez Twidosoft.
3	Sélectionnez le menu "Automate", puis "Sélectionner une connexion", et sélectionnez "My modem" (ou le nom que vous avez donné à votre connexion modem, voir "création d'une connexion" :) 
4	Connectez TwidoSoft

**Note :** Si vous voulez tout le temps utiliser votre connexion modem, dans le menu "fichier", "préférences", sélectionnez "my modem" (ou le nom que vous lui avez donné), ainsi, Twidosoft mémorisera cette préférence.

**Modes  
opératoires**

Le contrôleur Twido envoie la chaîne d'initialisation au modem connecté sous tension. Lorsqu'un modem est configuré dans l'application Twido, le contrôleur envoie d'abord une commande "FF" afin de savoir si le modem est connecté. Si le contrôleur reçoit une réponse, alors la chaîne d'initialisation est envoyée au modem.

## Appel Interne, Externe et International

Si vous communiquez avec un contrôleur Twido dans l'enceinte de votre entreprise, vous pouvez seulement utiliser l'extension de la ligne que vous devez composer, comme: 8445

Nom	Type de connexion	Phone	Timeout	Break timeout
COM1	Série	COM1	5000	20
COM4	Série	COM4	5000	20
My Modem 1	MODEM: TOSHIBA Internal V.90	8445	5000	20

Ajouter Modifier Supprimer OK

Si vous utilisez un standard interne pour composer les numéros de téléphone en dehors de votre entreprise et que vous devez faire un "0" ou un "9" avant le numéro de téléphone utilisez cette syntaxe : 0,0231858445 ou 9, 0231858445

Nom	Type de connexion	Phone	Timeout	Break timeout
COM1	Série	COM1	5000	20
COM4	Série	COM4	5000	20
My Modem 1	MODEM: TOSHIBA Internal V.90	0,0231858445	5000	20

Ajouter Modifier Supprimer OK

Pour les appels internationaux la syntaxe est : +19788699001 par exemple. Et si vous utilisez un standard : 0,+ 19788699001

Nom	Type de connexion	Phone	Timeout	Break timeout
COM1	Série	COM1	5000	20
COM4	Série	COM4	5000	20
My Modem 1	MODEM: TOSHIBA Internal V.90	0,+19788699	5000	20

Ajouter Modifier Supprimer OK

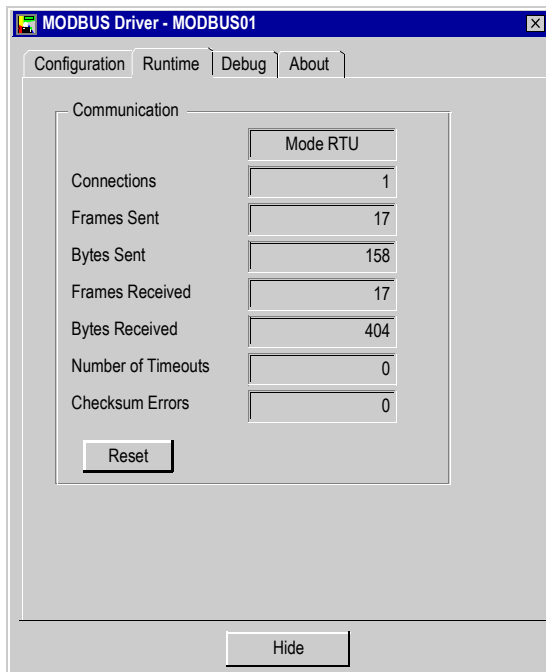
**Questions  
fréquemment  
posées**

Lorsque votre communication est établie depuis quelques minutes, vous pouvez avoir quelques erreurs de communication. Dans ce cas, vous devez ajuster les paramètres de communication.

Twidosoft utilise un driver modbus pour communiquer à travers des ports série ou des modems internes. Lorsque la communication a débuté, le driver modbus est visible dans la barre d'outils. Double cliquez sur l'icône du driver modbus pour ouvrir la fenêtre. Vous avez désormais accès aux paramètres du driver modbus, et l'onglet "runtime" vous donnera des informations sur les trames échangés avec le contrôleur à distance.

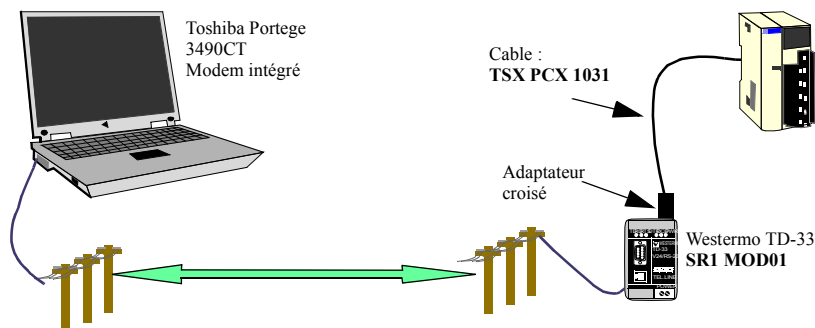
Si le "Nombre de timeouts" augmente ou est différent de 0, changez la valeur avec "Gestion des connexions" accessible sous Twidosoft, par le menu "Fichier" puis "préférences" "Gestion des connexions". Cliquez sur le champ "timeout", puis cliquez sur le bouton de modification et entrez une nouvelle valeur plus élevée. La valeur par défaut est "5000", en millième de secondes.

Essayez ensuite une nouvelle connexion. Ajustez la valeur jusqu'à ce que votre connexion devienne stable.



## Exemples

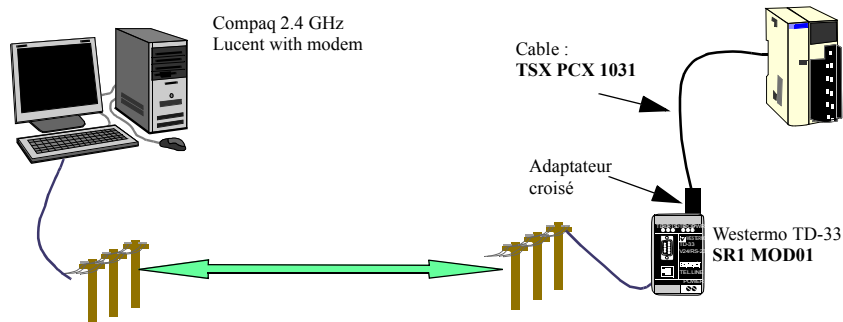
- **Exemple 1** : Twidosoft connecté à un TWD LMDA 20DRT (Windows 98 SE) .
  - PC : Toshiba portege 3490CT sous Windows 98,
  - Modem (interne au PC) : Toshiba internal V.90 modem,
  - Contrôleur Twido : TWD LMDA 20DRT version 2.0,
  - Modem (connecté au Twido) : Type Westermo TD-33 / V.90 référence SR1 MOD01 disponible sur le nouveau catalogue Twido (Septembre 03) (voir *Annexe 2, p. 113*),
  - Câble : TSX PCX 1031 connecté au port 1 de communication Twido, et un adaptateur: 9 pin male / 9 pin male afin de croiser Rx et Tx durant la connexion entre le modem Westermo et le contrôleur Twido (voir *Annexe 1, p. 112*). Vous pouvez également utiliser le câble TSX PCX 1130 (conversion RS485/232 et croisement Rx/Tx ).



Le premier test consiste à utiliser 2 lignes de téléphone analogiques, internes à l'entreprise, n'utilisant pas le numéro de téléphone complet, mais juste l'extension (C'est pourquoi il y a seulement 4 digits pour le numéro de téléphone de modem Toshiba V.90 interne).

Pour ce test, les paramètres de connexion (Twidosoft menu "préférences" puis "Gestion des connexions") étaient établis à leur valeur de défaut, soit timeout = 5000 et break timeout = 20.

- **Exemple 2** : Twidosoft connecté à TWD LMDA 20DRT (windows XP Pro)
  - PC : Compaq pentium 4, 2.4GHz,
  - Modem : Lucent Win modem, PCI bus,
  - Contrôleur Twido : TWD LMDA 20DRT version 2.0,
  - Modem (connecté au Twido) : Type WESTERMO TD-33 / V.90 référence SR1 MOD01 disponible dans le nouveau catalogue Twido (Septembre 03) (voir *Annexe 2, p. 113*),
  - Câble : TSX PCX 1031 connecté au port 1 de communication Twido, et un adaptateur : 9 pin male / 9 pin male afin de croiser Rx et Tx durant la connexion entre le modem Westermo et le contrôleur Twido (voir *Annexe 1, p. 112*). Vous pouvez également utiliser le câble TSX PCX 1130 (conversion RS485/232 et croisement Rx/Tx).

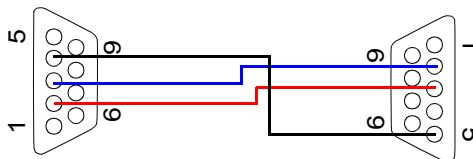


Le test consiste à utiliser deux lignes de téléphone analogiques, internes à l'entreprise, n'utilisant pas le numéro de téléphone complet, mais juste l'extension (C'est pourquoi il y a seulement 4 digits pour le numéro de téléphone du modem interne Toshiba V.90).

Pour ce test, les paramètres de connexion (Twidosoft, menu "préférences" puis "Gestion des connexions") étaient établis à leur valeur de défaut, soit timeout = 5000 et break timeout = 20.

## Annexe 1

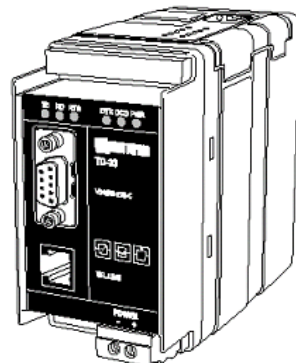
Adaptateur croisé pour le câble TSX PCX 1031 et modem Westermo TD-33 (SR1 MOD01) :



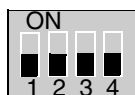


**Annexe 2**

Modem Westermo TD-33, référence Schneider **SR1 MOD01**. Ce modem gère 4 interrupteurs DIP, qui doivent tous être sur **OFF** :



Réglages usine



Utiliser la configuration prédéfinie (vitesse & format etc) Désactiver DTR Hotcall, Auto Band

**Annexe 3**

Modem Wavecom WMOD2B, référence Schneider **SR1 MOD02** double bande (900/1800Hz) :

**Annexe 4**

Références des produits utilisés dans ce document :

- Produit Twido : TWD LMDA 20DRT,
- Logiciel Twidosoft : TWD SPU 1002 V10M,
- Câble TSX PCX 1031,
- Câble TSX PCX 1130,
- Modem RTU : Westermo TD-33 / V90 **SR1 MOD01**,
- Modem GSM : Wavecom WMOD2B **SR1 MOD02**.


## Communications de liaison distante

### Introduction

La liaison distante est un bus maître/esclave à haut débit conçu pour assurer l'échange d'une petite quantité de données entre l'automate maître et un maximum de sept automates (esclaves) distants. Les données de l'application ou les données d'E/S sont transférées en fonction de la configuration des automates distants. Il est possible d'associer différents types d'automates, tels que des automates d'E/S distantes et des automates d'extension.

**Note :** L'automate maître contient les informations relatives au repère d'une E/S distante, mais il ne sait pas à quel automate précis correspond ce repère. Par conséquent, l'automate maître ne peut pas affirmer que toutes les entrées et sorties distantes utilisées dans l'application utilisateur existent réellement. Assurez-vous que cela soit le cas.

**Note :** Le bus d'E/S distantes et le protocole utilisé sont propriétaires et aucun périphérique tiers n'est autorisé sur le réseau.

	<b>ATTENTION</b>
	<b>FONCTIONNEMENT INATTENDU DU MATERIEL</b> <ul style="list-style-type: none"><li>• Assurez-vous qu'il existe un seul automate maître sur une liaison distante et que chaque esclave dispose d'un repère unique. Le non-respect de cette précaution risque d'altérer les données ou de générer des résultats inattendus et ambigus.</li><li>• Assurez-vous que tous les esclaves disposent d'un repère unique. Deux esclaves ne doivent pas avoir le même repère. Le non-respect de cette précaution risque d'altérer les données ou de générer des résultats inattendus et ambigus.</li></ul> <p><b>Le non-respect de ces précautions peut entraîner des lésions corporelles ou des dommages matériels.</b></p>

**Note :** La liaison distante nécessite une connexion EIA RS-485 et peut s'exécuter sur un seul port de communication à la fois.

## Configuration matérielle

Une liaison distante doit utiliser un port EIA RS-485 à 3 fils minimum. Il est possible de la configurer afin d'utiliser le premier port ou un deuxième port optionnel existant.

**Note :** Un seul port de communication à la fois peut être configuré en tant que liaison distante.

Le tableau suivant répertorie les périphériques utilisables :

Automate	Port	Caractéristiques
TWDLCA10/16/24DRF, TWDLCA40DRF, TWDLMDA20/40DUK, TWDLMDA20/40DTK, TWDLMDA20DRT	1	Base automate équipé d'un port EIA RS-485 à 3 fils avec un connecteur mini DIN.
TWDNOZ485D	2	Module de communication équipé d'un port EIA RS-485 à 3 fils avec un connecteur mini DIN. <b>Remarque :</b> Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion d'afficheur.
TWDNOZ485T	2	Module de communication équipé d'un port EIA RS-485 à 3 fils avec un bornier. <b>Remarque :</b> Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion d'afficheur.
TWDNAC485D	2	Adaptateur de communication équipé d'un port EIA RS-485 à 3 fils avec un connecteur mini DIN. <b>Remarque :</b> Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion d'afficheur.
TWDNAC485T	2	Adaptateur de communication équipé d'un port EIA RS-485 à 3 fils avec un bornier. <b>Remarque :</b> Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion d'afficheur.
TWDXCPODM	2	Module d'expansion de l'afficheur équipé d'un port EIA RS-485 à 3 fils avec un connecteur mini DIN ou d'un port EIA RS-485 à 3 fils avec un bornier. <b>Remarque :</b> Ce module n'est disponible que pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion de communication.

**Note :** La vérification de la présence du port 2 et de sa configuration (RS232 ou RS485) est uniquement réalisée lors de la mise sous tension ou de la réinitialisation par le microprogramme de l'automate.

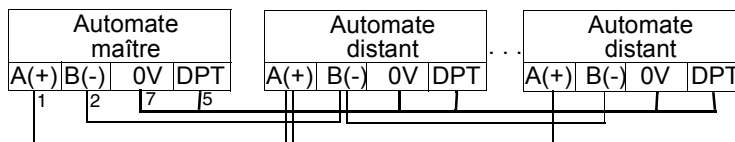
## Connexion de câbles à chaque périphérique

**Note :** Le signal DPT sur la broche 5 doit être relié au 0V sur la broche 7 afin de signaler l'utilisation de communications de liaison distante. Lorsque ce signal n'est pas relié à la terre, l'automate Twido maître ou esclave est défini par défaut dans un mode dans lequel des tentatives d'établir des communications avec TwidoSoft s'effectuent.

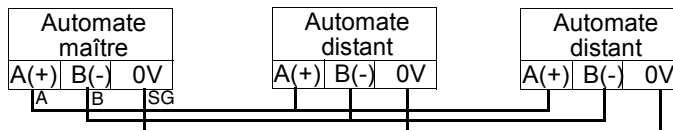
**Note :** La connexion DPT à 0V n'est nécessaire qu'en cas de connexion à une base automate sur le port 1.

Les connexions de câbles effectuées à chaque périphérique sont représentées ci-dessous.

### Connexion mini DIN




### Connexion bornier



**Configuration  
logicielle**

Un seul automate maître doit être défini sur la liaison distante. En outre, chaque automate distant doit conserver un repère esclave unique. L'utilisation de repères identiques par plusieurs maîtres ou esclaves risque d'altérer des transmissions ou de créer des ambiguïtés.

	<b>ATTENTION</b>
	<p><b>Fonctionnement inattendu du matériel</b></p> <p>Assurez-vous qu'il existe un seul automate maître sur une liaison distante et que chaque esclave dispose d'un repère unique. Le non-respect de cette précaution risque d'altérer les données ou de générer des résultats inattendus et ambigus.</p> <p><b>Le non-respect de ces précautions peut entraîner des lésions corporelles ou des dommages matériels.</b></p>

**Configuration de  
l'automate maître**

Configurez l'automate maître à l'aide de TwidoSoft pour gérer un réseau de liaison distante constitué au maximum de sept automates distants. Ces sept automates distants peuvent être configurés en tant qu'E/S distantes ou automates d'extension. Le repère du maître configuré à l'aide de TwidoSoft correspond au repère 0. Pour configurer un automate en maître vous devez configurer le port 1 ou le port 2 en liaison distante et choisir le repère 0 (Maître) à l'aide de TwidoSoft. Puis à l'aide de la fenêtre "Ajouter un automate distant", vous déterminez les automates esclaves soit en E/S distantes, soit en automates d'extension ainsi que leur repère.

**Configuration de l'automate distant**

La configuration d'un automate distant s'effectue à l'aide de TwidoSoft en configurant le port 1 ou 2 en liaison distante et ou en lui affectant un repère entre 1 et 7.

Le tableau suivant résume les différences et les contraintes de chacun de ces types de configuration d'automate distant.

Type	Programme d'application	Accès aux données
E/S distantes	Non  Pas même une simple instruction "END" Le mode RUN dépend de celui du maître.	%I et %Q  Seule l'E/S locale de l'automate distant est accessible (et non son extension d'E/S).
Automate d'extension	Oui  Le mode RUN est indépendant de celui du maître.	%INW et %QNW  Il est possible de transmettre un maximum de quatre mots d'entrée et quatre mots de sortie vers et depuis chaque extension.

**Synchronisation  
de scrutation de  
l'automate  
distant**

Le cycle de mise à jour de la liaison distante n'est pas synchronisé avec la scrutation de l'automate maître. Les communications avec les automates distants sont déclenchées par interruption et se produisent en tant que tâches en arrière-plan, en parallèle avec l'exécution de la scrutation de l'automate maître. A la fin du cycle de scrutation, les valeurs les plus récentes sont lues dans les données d'application à utiliser pour la prochaine exécution de programme. Ce traitement est le même pour les automates d'E/S distantes et d'extension.

Tous les automates peuvent vérifier l'activité de la liaison générale à l'aide du bit système %S111. Mais pour accomplir la synchronisation, un automate maître ou d'extension doit utiliser le bit système %S110. Ce bit est mis à 1 une fois qu'un cycle de mise à jour complet s'est déroulé. Le programme d'application est responsable de sa remise à 0.

Le maître peut activer ou désactiver la liaison distante à l'aide du bit système %S112. Les automates peuvent contrôler la configuration et le bon fonctionnement de la liaison distante à l'aide de %S113. Le signal DPT sur le port 1 (utilisé pour déterminer si TwidoSoft est connecté) est détecté et signalé sur %S100.

Le tableau suivant résume toutes ces informations.

Bit système	Etat	Indication
%S100	0	maître/esclave : DPT inactif (câble TwidoSoft NON connecté)
	1	maître/esclave : DPT actif (câble TwidoSoft connecté)
%S110	0	maître/esclave : mis à 0 par l'application
	1	maître : tous les échanges de liaison distante effectués (E/S distantes uniquement) esclave : échange avec maître effectué
%S111	0	maître : échange de liaison distante unique effectué esclave : échange de liaison distante unique détecté
	1	maître : échange de liaison distante unique en cours esclave : échange de liaison distante unique détecté
%S112	0	maître : liaison distante désactivée
	1	maître : liaison distante activée
%S113	0	maître/esclave : configuration/fonctionnement de la liaison distante OK
	1	maître : erreur de configuration/fonctionnement de la liaison distante esclave : erreur de fonctionnement de la liaison distante

### Redémarrage de l'automate maître

Lorsqu'un automate maître redémarre, l'un des événements suivants se produit :

- Un démarrage à froid (%S0 = 1) force la réinitialisation des communications.
- Un démarrage à chaud (%S1 = 1) force la réinitialisation des communications.
- En mode Stop, le maître continue à communiquer avec les esclaves.

### Redémarrage de l'automate esclave

Lorsqu'un automate esclave redémarre, l'un des événements suivants se produit :

- Un démarrage à froid (%S0 = 1) force la réinitialisation des communications.
- Un démarrage à chaud (%S1 = 1) force la réinitialisation des communications.
- En mode Stop, l'esclave continue de communiquer avec le maître. Si le maître indique un état Stop :
  - Les E/S distantes appliquent un état Stop.
  - L'automate d'extension continue dans son état actuel.

### Arrêt de l'automate maître

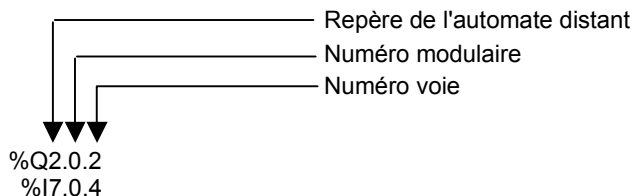
Lorsque l'automate maître passe en Stop, tous les périphériques esclaves continuent de communiquer avec le maître. Lorsque le maître indique qu'un arrêt est requis, un automate d'E/S distantes s'arrête, mais les automates d'extension continuent dans leur état courant d'exécution et d'arrêt.

### Accès aux données E/S distantes

L'automate distant configuré en tant qu'E/S distantes ne possède, ni n'exécute son propre programme d'application. Les entrées et sorties TOR de base de l'automate distant sont une simple extension de celles de l'automate maître. L'application doit uniquement utiliser le mécanisme de repérage complet à trois chiffres fourni.

**Note :** Le numéro de module est toujours zéro pour les E/S distantes.

Illustration

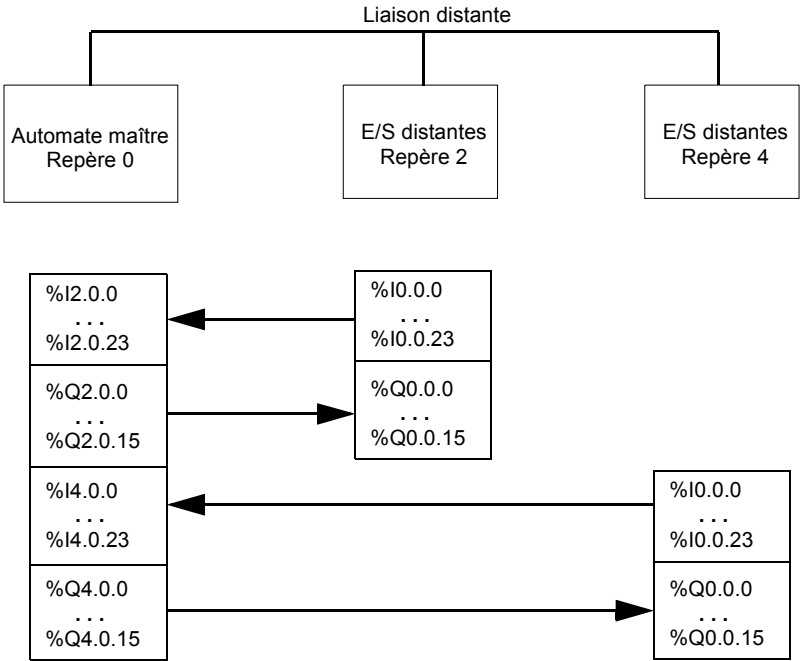


Pour communiquer avec les E/S distantes, l'automate maître utilise la notation d'entrée et sortie standard %I et %Q. Pour accéder au troisième bit de sortie de l'E/S distante configurée au repère 2, on utilise l'instruction %Q2.0.2. De même, pour lire le cinquième bit d'entrée de l'E/S distante configurée au repère 7, on utilise l'instruction %I7.0.4

**Note :** L'accès du maître est restreint aux E/S TOR appartenant aux E/S locales de l'automate distant. Aucune E/S analogique ou d'expansion ne peut être transférée, hormis en cas d'utilisation de communications d'extension.

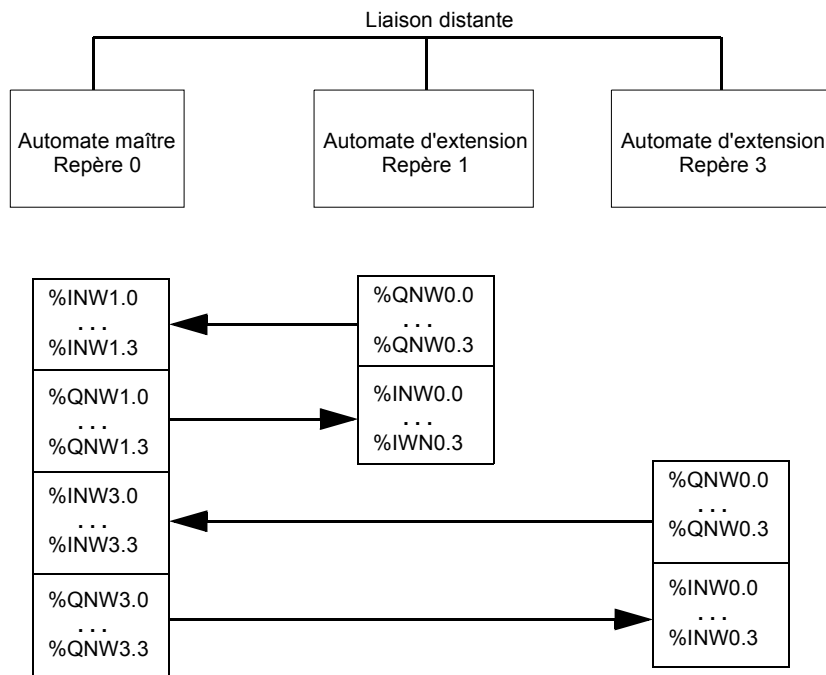


Illustration



### Accès aux données de l'automate d'extension

Pour communiquer avec des automates d'extension, le maître utilise les mots réseau %INW et %QNW afin d'échanger des données. Chaque extension du réseau est accessible par son repère distant "j" à l'aide de mots %INWj.k et %QNWj.k. Chaque automate d'extension du réseau utilise %INW0.0 à %INW0.3 et %QNW0.0 à %QNW0.3 pour accéder aux données situées sur le maître. Les mots réseau sont automatiquement mis à jour lorsque les automates sont en mode Run ou Stop. L'exemple suivant illustre l'échange d'un maître avec deux automates d'extension configurés.



Il n'existe aucune remise de messages de poste à poste au sein de la liaison distante. Il est possible d'utiliser le programme application du maître pour gérer les mots réseaux, afin de transférer des informations entre des automates distants, qui utilisent alors le maître en tant que passerelle.

## Informations d'état

Outre les bits système décrits précédemment, le maître conserve l'état de présence et de configuration des automates distants. Cette action s'effectue dans les mots systèmes %SW111 et %SW113. L'automate maître ou l'automate distant peut obtenir la valeur de la dernière erreur survenue pendant la communication sur la liaison distante dans le mot système %SW112.

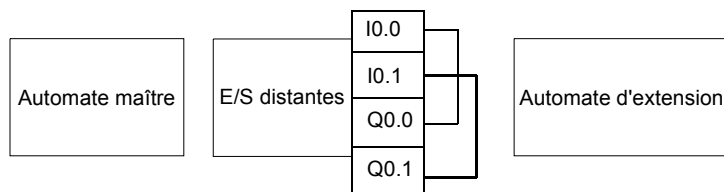
Mots système	Utilisation	
%SW111	Etat de la liaison distante : deux bits pour chaque automate distant (maître uniquement)	
	x0-6	0 - automate distant 1-7 absent
		1 - automate distant 1-7 présent
	x8-14	0 - E/S distante détectée sur l'automate distant 1-7
		1 - automate d'extension détecté sur l'automate distant 1-7
%SW112	Code d'erreur de configuration ou de fonctionnement de la liaison distante	
		0 - opérations réussies
		1 - expiration du délai (esclave)
		2 - erreur de checksum détectée (esclave)
		3 - incohérence de configuration (esclave)
%SW113	Configuration de la liaison distante : deux bits pour chaque automate distant (maître uniquement)	
	x0-6	0 - automate distant 1-7 non configuré
		1 - automate distant 1-7 configuré
	x8-14	0 - E/S distante configurée en tant qu'automate distant 1-7
		1 - automate d'extension configuré en tant qu'automate distant 1-7

## Exemple de liaison distante

Pour configurer une liaison distante, procédez comme suit :

1. Configurez le matériel.
2. Câblez les automates.
3. Connectez le câble de communication entre le PC et les automates.
4. Configurez le logiciel.
5. Ecrivez une application.

Les illustrations suivantes représentent une utilisation de la liaison distante avec les E/S distantes et un automate d'extension.

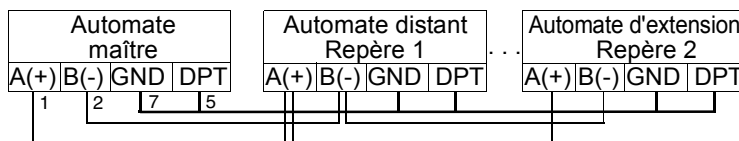
**Etape 1 : Configuration du matériel :**

La configuration matérielle comprend trois bases automates de tout type. Le port 1 est utilisé selon deux modes de communication. L'un des modes permet de configurer et de transférer le programme d'application à l'aide de TwidoSoft. Le second mode est destiné au réseau de liaison distante. Si un port 2 optionnel est disponible sur l'un des automates, il est possible de l'utiliser, mais tout automate ne gère qu'une seule liaison distante.

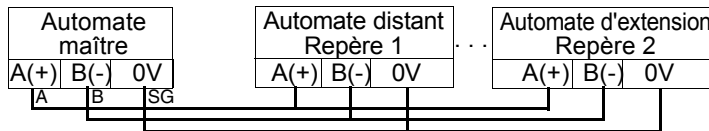
**Note :** Dans cet exemple, les deux premières entrées sur les E/S distantes sont câblées sur les deux premières sorties.

**Etape 2 : Câblage des automates :**

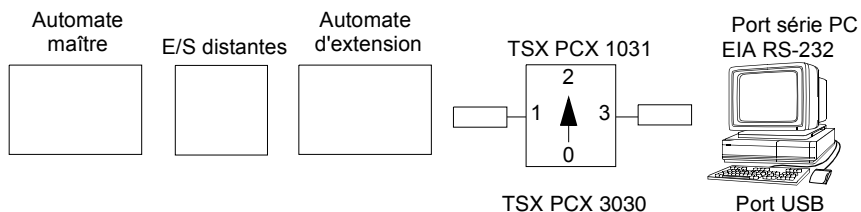
Connexion mini DIN



Connexion bornier



Connectez les câbles des signaux A(+) et B(-) ensemble. Sur chaque automate, le signal DPT est relié à la terre. Bien que la mise à la terre du signal ne soit pas obligatoire pour une utilisation avec une liaison distante sur le port 2 (cartouche ou module de communication optionnels), il s'agit d'une bonne habitude à prendre.

**Etape 3 : Connexion du câble de communication entre le PC et les automates :**

Le câble de programmation multifonctions TSX PCX 1031 ou TSX PCX 3030 est utilisé pour communiquer avec chacune des trois bases automates. Assurez-vous que le commutateur du câble est en position 2. Afin de programmer chaque automate, il est nécessaire d'établir une communication point à point avec chaque automate. Pour établir cette communication : connectez-vous au port 1 du premier automate, transférez la configuration et les données de l'application, puis mettez l'automate en Run. Répétez cette procédure pour chaque automate.

**Note :** Il est nécessaire de déplacer le câble après chaque configuration d'automate et transfert d'application.

**Etape 4 : Configuration du logiciel :**

Chacun des trois automates utilise TwidoSoft pour créer une configuration, et le cas échéant, le programme d'application.

Pour l'automate maître, éditez le paramétrage des communications de l'automate afin de régler le protocole sur "Liaison distante" et le repère sur "0 (Maître)".

**Paramètre Comm. de l'automate**

Type : Liaison distante  
Repère : 0 (maître)

Configurez l'automate distant sur le maître en ajoutant une "E/S distante" au repère "1" et un "Automate d'extension" au repère "2".

**Ajouter automates distants**

Utilisation automate : E/S distantes  
Adresse distante : 1

Utilisation automate : Automate d'extension  
Adresse distante : 2

Pour l'automate configuré en tant qu'E/S distantes, vérifiez que le paramétrage des communications de l'automate est réglé sur "Liaison distante" et sur le repère "1".

**Paramètre Comm. de l'automate**

Type : Liaison distante  
Repère : 1

Pour l'automate configuré en tant qu'extension, vérifiez que la configuration de la communication de l'automate est réglée sur "Liaison distante" et sur le repère "2".

**Paramètre Comm. de l'automate**

Type : Liaison distante  
Repère : 2

**Etape 5 : Ecriture des applications :**

Pour l'automate maître, écrivez le code du programme d'application suivant :

```
LD 1

[%MW0 := %MW0 +1]
[%QNW2.0 := %MW0]
[%MW1 := %INW2.0]

LD %I0.0
ST %Q1.00.0
LD %I1.0.0
ST %Q0.0

LD %I0.1
ST %Q1.0.1
LD %I1.0.1
ST %Q0.1
```

Pour l'automate configuré en tant qu'E/S distantes, n'écrivez pas de programme d'application.

Pour l'automate configuré en tant qu'extension, écrivez l'application suivante :

```
LD 1
[%QNW0.0 := %INW0.0]
```

Dans cet exemple, l'application maître incrémente un mot mémoire interne et le communique à l'automate d'extension à l'aide d'un seul mot réseau. L'automate d'extension prend le mot reçu du maître et le renvoie. Dans le maître, un mot mémoire différent reçoit et stocke cette transmission.

Pour communiquer avec l'automate d'E/S distantes, le maître envoie ses entrées locales aux sorties des E/S distantes. A l'aide de la connexion E/S externe des E/S distantes, les signaux sont renvoyés et récupérés par le maître.

---

## Communications ASCII

### Introduction

Le protocole ASCII offre aux automates Twido un protocole de mode caractère semi-duplex simple permettant d'émettre et/ou de recevoir des données à l'aide d'un seul périphérique. Ce protocole est pris en charge à l'aide de l'instruction EXCHx et géré à l'aide du bloc fonction %MSGx.

Les trois types de communications suivants sont possibles à l'aide du protocole ASCII :

- Emission seule
- Emission/réception
- Réception seule

La taille maximale des trames émises et/ou reçues à l'aide de l'instruction EXCHx s'élève à 256 octets.

### Configuration matérielle

Il est possible d'établir une liaison ASCII (voir les bits systèmes %S103 et %S104 (Voir *Bits système (%S)*, p. 546)) sur le port EIA RS-232 ou EIA RS-485 et de l'exécuter simultanément sur deux ports de communication au maximum.

Le tableau suivant répertorie les périphériques utilisables :

Automate	Port	Caractéristiques
TWDLCA10/16/24DRF, TWDLCA40DRF, TWDLMDA20/40DUK, TWDLMDA20DRT	1	Base automate équipée d'un port EIA RS-485 à 3 fils avec un connecteur mini DIN.
TWDNOZ232D	2	Module de communication équipé d'un port EIA RS-232 à 3 fils avec un connecteur mini DIN. <b>Remarque</b> : Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Afficheur.
TWDNOZ485D	2	Module de communication équipé d'un port EIA RS-485 à 3 fils avec un connecteur mini DIN. <b>Remarque</b> : Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Afficheur.
TWDNOZ485T	2	Module de communication équipé d'un port EIA RS-485 à 3 fils avec un bornier. <b>Remarque</b> : Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion d'afficheur.

Automate	Port	Caractéristiques
TWDNAC232D	2	Adaptateur de communication équipé d'un port EIA RS-232 à 3 fils avec un connecteur mini DIN. <b>Remarque :</b> Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDNAC485D	2	Adaptateur de communication équipé d'un port EIA RS-485 à 3 fils avec un connecteur mini DIN. <b>Remarque :</b> Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDNAC485T	2	Adaptateur de communication équipé d'un port EIA RS-485 à 3 fils avec un bornier. <b>Remarque :</b> Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDXCPODM	2	Module d'expansion de l'afficheur équipé d'un port EIA RS-232 à 3 fils avec un connecteur mini DIN, d'un port EIA RS-485 à 3 fils avec un connecteur mini DIN, et d'un port EIA RS-485 à 3 fils avec un bornier. <b>Remarque :</b> Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Communication.

**Note :** La vérification de la présence du port 2 et de sa configuration (RS232 ou RS485) est uniquement réalisée lors de la mise sous tension ou de la réinitialisation par le microprogramme de l'automate.

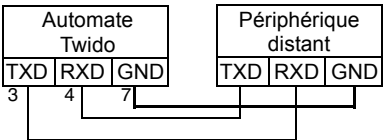


**Câblage nominal** Les connexions de câble nominal sont représentées ci-dessous pour les types EIA RS-232 et EIA RS-485.

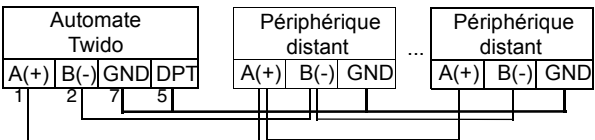
**Note :** Si le port 1 est utilisé sur l'automate Twido, le signal DPT sur la broche 5 doit être relié au 0V de la broche 7. Ce signal indique à l'automate Twido que les communications via le port 1 relèvent du protocole ASCII et non du protocole utilisé pour communiquer avec le logiciel TwidoSoft.

Les connexions de câbles de chaque périphérique sont représentées ci-dessous.  
Connexion mini DIN

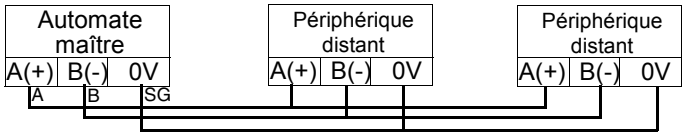
Câble EIA RS-232



Câble EIA RS-485



Connexion bornier



**Configuration logicielle**

Pour configurer l'automate afin d'utiliser une liaison série pour envoyer et recevoir des caractères à l'aide du protocole ASCII, procédez comme suit :

Etape	Description
1	Configurez le port série pour le protocole ASCII à l'aide de TwidoSoft.
2	Créez dans votre application une table d'émission/réception qui sera utilisée par l'instruction EXCHx.

**Configuration du port**

Un automate Twido peut utiliser son port 1 principal ou un port 2 configuré en option pour utiliser le protocole ASCII. Pour configurer un port série pour le protocole ASCII :

Etape	Action
1	Définissez tous les modules ou adaptateurs de communication supplémentaires configurés sur la base.
2	Cliquez avec le bouton droit sur le port, puis cliquez sur Paramétrer les communications de l'automate et modifiez le type du port série sur "ASCII".
3	Définissez les paramètres de communication associés.

**Configuration de la table d'émission/réception du mode ASCII**

La taille maximale des trames émises et/ou reçues s'élève à 256 octets. La table de mots associée à l'instruction EXCHx se compose des tables de contrôle d'émission et de réception.

	Octet de poids fort	Octet de poids faible
Table de contrôle	Commande	Longueur (émission/réception)
	Réservés (0)	Réservés (0)
Table d'émission	Octet 1 émis	Octet 2 émis
	...	...
	...	Octet n émis
	Octet n+1 émis	
Table de réception	Octet 1 reçu	Octet 2 reçu
	...	...
	...	Octet p reçu
	Octet p+1 reçu	

**Table de contrôle**

L'octet **Longueur** contient la longueur de la table d'émission en octets (250 max), qui est écrasée par le nombre de caractères reçus à la fin de la réception, si la réception est demandée.

L'octet **Commande** doit contenir l'un des éléments suivants :

- 0: Emission seule
- 1: Emission/réception
- 2: Réception seule

**Tables de  
d'émission/  
réception**

En mode Emission seule, les tables de contrôle et d'émission sont renseignées avant l'exécution de l'instruction EXCHx ; elles peuvent être de type %KW ou %MW. Aucun espace n'est requis pour la réception des caractères en mode Emission seule. Une fois que tous les octets ont été émis %MSGx.D est réglé sur 1 ; il est alors possible d'exécuter une nouvelle instruction EXCHx.

En mode Emission/Réception, les tables de contrôle et d'émission sont renseignées avant l'exécution de l'instruction EXCHx ; elles doivent être de type %MW. Un espace prévu pour un maximum de 256 octets de réception est requis à la fin de la table d'émission. Une fois que tous les octets ont été émis, l'automate Twido passe en mode de réception et est prêt à recevoir des octets.

En mode Réception seule, la table de contrôle est renseignée avant l'exécution de l'instruction EXCHx ; elle doit être de type %MW. Un espace prévu pour un maximum de 256 octets de réception est requis à la fin de la table de contrôle.

L'automate Twido passe immédiatement en mode de réception et est prêt à recevoir des octets.

La réception est terminée une fois que l'octet de fin de trame a été reçu ou lorsque la table de réception est pleine. Dans ce cas, une erreur apparaît (débordement de la table de réception) dans le mot %SW63 et %SW64. Si un délai différent de zéro est configuré, la réception se termine lorsque ce délai est écoulé. Si vous sélectionnez une valeur de délai égale à zéro, il n'existe aucun délai de réception. Par conséquent, pour arrêter la réception, activez l'entrée %MSGx.R.

**Echange de  
messages**

Le langage propose deux services pour la communication :

- **Instruction EXCHx** : pour émettre/recevoir des messages.
- **Bloc fonction %MSGx** : pour contrôler les échanges de messages.

L'automate Twido utilise le protocole configuré pour ce port lors du traitement d'une instruction EXCHx.

**Note** : Il est possible de configurer chaque port de communication pour différents protocoles ou pour le même protocole. Pour accéder à l'instruction EXCHx ou au bloc fonction %MSGx de chaque port de communication, il suffit d'ajouter le numéro du port (1 ou 2).

**Instruction  
EXCHx**

L'instruction EXCHx permet à l'automate Twido d'envoyer et/ou de recevoir des informations vers/depuis des périphériques ASCII. L'utilisateur définit une table de mots (%MWi:L ou %KWi:L) contenant des informations de contrôle, ainsi que les données à envoyer et/ou à recevoir (jusqu'à 256 octets en émission et/ou réception). La description du format de la table de mots a été donnée précédemment.

Un échange de messages s'effectue à l'aide de l'instruction EXCHx.

Syntaxe : [EXCHx %MWi:L]

où : x = numéro du port (1 ou 2)

L = nombre de mots dans les tables de mots de commande,  
d'émission et de réception

L'automate Twido doit terminer l'échange de la première instruction EXCHx avant de pouvoir en lancer une deuxième. Il est nécessaire d'utiliser le bloc fonction %MSGx lors de l'envoi de plusieurs messages.

Le traitement de l'instruction par liste EXCHx se produit immédiatement, en sachant que toutes les émissions sont démarrées sous contrôle d'interruptions (la réception des données est également sous contrôle d'interruptions), ce qui est considéré comme un traitement en arrière-plan.

---

**Bloc fonction  
%MSGx**

L'utilisation du bloc fonction %MSGx est facultative ; il permet de gérer des échanges de données. Le bloc fonction %MSGx remplit trois fonctions :

- **Vérification des erreurs de communications**

La recherche d'erreurs permet de vérifier que le paramètre L (longueur de la table de mots) programmée à l'aide de l'instruction EXCHx est suffisamment grand pour contenir la longueur du message à envoyer. Celle-ci est comparée à la longueur programmée dans l'octet de poids faible du premier mot de la table de mots.

- **Coordination de plusieurs messages**

Pour garantir la coordination lors de l'envoi de plusieurs messages, le bloc fonction %MSGx fournit les informations requises pour déterminer le moment où l'émission du message précédent est terminée.

- **Emission de messages prioritaires**

Le bloc fonction %MSGx vous permet de suspendre l'émission d'un message afin d'envoyer un message plus urgent.

Le bloc fonction %MSGx dispose d'une entrée et de deux sorties qui lui sont associées :

Entrée/Sortie	Définition	Description
R	Entrée RAZ	A l'état 1 : réinitialise la communication ou le bloc (%MSGx.E = 0 et %MSGx.D = 1).
%MSGx.D	Communication terminée	0 : requête en cours. 1 : communication terminée en cas de fin de transmission, de réception du caractère de fin, d'erreur ou de réinitialisation du bloc.
%MSGx.E	Erreur	0 : longueur du message OK et liaison OK. 1 : en cas de mauvaise commande, de table configurée de manière incorrecte, de mauvais caractère reçu (débit, parité, etc.) ou de saturation de la table de réception.

## Limitations

Il est important de garder à l'esprit les limitations suivantes :

- La disponibilité et le type du port 2 (voir %SW7) sont uniquement contrôlés lors de la mise sous tension ou de la réinitialisation.
- Tout message en cours de traitement sur le port 1 est abandonné lorsque TwidoSoft est connecté.
- Il est impossible de traiter EXCHx ou %MSG sur un port configuré en tant que liaison distante.
- EXCHx abandonne le traitement Modbus esclave actif
- Le traitement des instructions EXCHx ne fait pas l'objet d'une nouvelle tentative en cas d'erreur.
- Il est possible d'utiliser l'entrée RAZ pour annuler le traitement de la réception d'une instruction EXCHx.
- Il est possible de configurer des instructions EXCHx avec un délai d'annulation de réception.
- Les messages multiples sont contrôlés via %MSGx.D.

**Erreurs et conditions de fonctionnement**

Si une erreur se produit lors de l'utilisation de l'instruction EXCHx, les bits %MSGx.D et %MSGx.E sont réglés sur 1, le mot système %SW63 contient le code d'erreur du port 1 et %SW64 le code d'erreur du port 2.

Mots système	Utilisation
%SW63	Code d'erreur EXCH1 : 0 - opération réussie 1 - nombre d'octets à émettre trop important (> 250) 2 - table d'émission trop petite 3 - table de mots trop petite 4 - débordement de la table de réception 5 - délai écoulé 6 - erreur d'émission 7 - mauvaise commande dans la table 8 - port sélectionné non configuré/disponible 9 - erreur de réception 10 - impossible d'utiliser %KW en cas de réception 11 - décalage d'émission plus important que la table d'émission 12 - décalage de réception plus important que la table de réception 13 - interruption du traitement EXCH par l'automate
%SW64	Code d'erreur EXCH2 : Voir %SW63.

**Conséquence du redémarrage de l'automate sur la communication**

Lorsqu'un automate redémarre, l'un des événements suivants se produit :

- Un démarrage à froid (%S0 = 1) force la réinitialisation des communications.
- Un démarrage à chaud (%S1 = 1) force la réinitialisation des communications.
- En mode Stop, l'automate arrête toutes les communications ASCII.

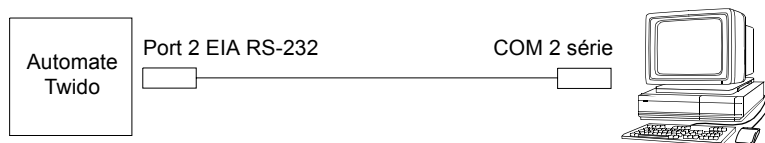
## Exemple de liaison ASCII

Pour configurer une liaison ASCII, procédez comme suit :

1. Configurez le matériel.
2. Connectez le câble de communication ASCII.
3. Configurez le port.
4. Ecrivez une application.
5. Initialisez l'éditeur de tables d'animation.

L'illustration suivante représente l'utilisation de la communication ASCII à l'aide d'un émulateur de terminal sur un PC.

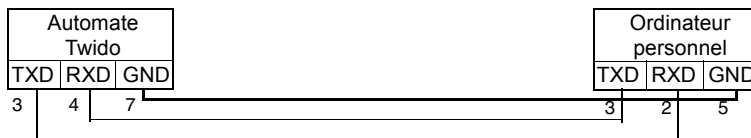
### Etape 1 : Configuration du matériel :



La configuration matérielle comporte deux liaisons séries entre le PC et un automate Twido doté d'un port 2 EIA RS-232 optionnel. Sur un automate modulaire, le port 2 optionnel correspond à TWDNOZ232D ou à TWDNAC232D dans le module TWDXCPODM. Sur l'automate compact, le port 2 optionnel est un port TWDNAC232D.

Pour configurer l'automate, connectez le câble TSXPCX1031 (non illustré) au port 1 de l'automate Twido. Connectez ensuite le câble au port COM 1 du PC. Vérifiez que le commutateur est en position 2. Enfin, connectez le port COM 2 du PC au port 2 EIA RS-232 de l'automate Twido. Le schéma de câblage est présenté à l'étape suivante.

### Etape 2 : Schéma de câblage de communication ASCII (EIA RS-232) :



Le nombre minimum de fils utilisé dans un câble de communication ASCII est 3. Croisez les signaux d'émission et de réception.

**Note :** A l'extrémité PC du câble, des connexions supplémentaires (telles que DTR et DSR) peuvent être nécessaires afin de satisfaire le protocole de transmission. Aucune connexion supplémentaire n'est requise pour l'automate Twido.

**Etape 3 : Configuration du port :**

Matériel -> Ajouter une option TWDNOZ232D
--

Matériel => Paramétrer paramètre Comm. Réglage
--

Port :	2
Type :	ASCII
Débit :	19 200
Données :	8 bits
Parité :	Aucune
Arrêt :	1 bit
Fin de trame :	65
Délai de réponse :	100 x 100 ms

Emulateur de terminal sur un PC
---------------------------------

Port :	COM2
Débit :	19 200
Données :	8 bits
Parité :	Aucune
Arrêt :	1 bit
Contrôle de flux :	Aucune

Utilisez une simple application d'émulateur de terminal sur le PC pour configurer le port COM2 et pour garantir l'absence de contrôle de flux.

Utilisez TwidoSoft pour configurer le port de l'automate. En premier lieu, configurez l'option matérielle. Dans cet exemple, le port TWDNOZ232D est ajouté à la base automate modulaire.

En second lieu, initialisez le paramétrage de la communication de l'automate à l'aide des mêmes paramètres que ceux de l'émulateur de terminal sur le PC. Dans cet exemple, la lettre majuscule "A" est choisie comme caractère de "fin de trame", afin de terminer la réception de caractère. Un délai de dix secondes est choisi pour le paramètre Délai de réponse. Un seul de ces deux paramètres sera utilisé, selon celui qui se produira en premier.

**Etape 4 : Ecriture d'une application :**

LD 1
[%MW10 := 16#0104]
[%MW11 := 16#0000]
[%MW12 := 16#4F4B]
[%MW13 := 16#0A0D]
LD 1
AND %MSG2.D
[EXCH2 %MW10:8]
LD %MSG2.E
ST %Q0.0
END

Utilisez TwidoSoft pour créer un programme d'application en trois temps. Tout d'abord, initialisez la table de contrôle et la table d'émission pour utiliser l'instruction EXCH. Dans cet exemple, une commande est configurée pour à la fois envoyer et recevoir des données. La quantité de données à envoyer est réglée sur quatre octets et initialisée sur les caractères : "O", "K", CR et LF.



Vérifiez ensuite le bit d'état de communication associé à %MSG2 et exécutez l'instruction EXCH2 uniquement si le port est prêt. Une valeur de 8 mots est spécifiée pour l'instruction EXCH2. Il existe deux mots de commande (%MW10 et %MW11), deux mots à utiliser pour les informations d'émission (%MW12 et %MW13) et quatre mots pour recevoir des données (%MW14 à %MW17). Finalement, l'état d'erreur du %MSG2 est détecté et stocké sur le premier bit de sortie sur l'E/S de la base automate locale. Vous pouvez également effectuer à l'aide de %SW64 une recherche d'erreurs supplémentaire pour rendre celle-ci plus précise.

**Etape 5 :** Initialisation de l'éditeur de tables d'animation :

Format courant conservé du repère		
1 %MW10	0104	Hexadécimal
2 %MW11	0000	Hexadécimal
3 %MW12	4F4B	Hexadécimal
4 %MW13	0A0D	Hexadécimal
5 %MW14	TW	ASCII
6 %MW15	ID	ASCII
7 %MW16	O	ASCII
8 %MW17	A	ASCII

L'étape finale consiste à télécharger cette application d'automate et à l'exécuter. Initialisez l'éditeur de tables d'animation pour animer et afficher les mots %MW10 à %MW17. Sur l'émulateur de terminal, les caractères "O"- "K"-CR-LF s'affichent. Les caractères "O"- "K"-CR-LF peuvent s'afficher autant de fois que le délai de réponse du bloc EXCH s'est écoulé et qu'un nouveau bloc EXCH a été lancé. Sur l'émulateur de terminal, tapez "T"- "W"- "I"- "D"- "O"- " "- "A". Ces informations sont échangées avec l'automate Twido et s'affichent dans l'éditeur de tables d'animation.

## Communications Modbus

### Introduction

Le protocole Modbus est un protocole maître-esclave qui permet à un seul et unique maître de demander des réponses à des esclaves ou d'agir en fonction de la requête. Le maître peut s'adresser aux esclaves individuellement ou envoyer un message de diffusion générale à tous les esclaves. Les esclaves renvoient un message (réponse) aux requêtes qui leur sont adressées individuellement. Les réponses aux requêtes de diffusion générale du maître ne sont pas renvoyées.

### Configuration matérielle

Il est possible d'établir une liaison Modbus sur le port EIA RS-232 ou EIA RS-485 et de l'exécuter simultanément sur deux ports de communication au maximum. Chaque port peut obtenir sa propre adresse Modbus, en utilisant le bit système %S101 et les mots système %SW101 et %SW102 (Voir *Bits système (%S)*, p. 546). (Voir aussi *Mots système (%SW)*, p. 555)

Le tableau suivant répertorie les périphériques qui peuvent être utilisés :

Automate	Port	Caractéristiques
TWDLCA10/16/24DRF, TWDLCA40DRF, TWDLMDA20/40DUK, TWDLMDA20DRT	1	Base automate prenant en charge un port EIA RS-485 à 3 fils avec un connecteur mini DIN.
TWDNOZ232D	2	Module de communication équipé d'un port EIA RS-232 à 3 fils avec un connecteur mini DIN. <b>Remarque :</b> Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Afficheur.
TWDNOZ485D	2	Module de communication équipé d'un port EIA RS-485 à 3 fils avec un connecteur mini DIN. <b>Remarque :</b> Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion Afficheur.
TWDNOZ485T	2	Module de communication équipé d'un port EIA RS-485 à 3 fils avec un bornier. <b>Remarque :</b> Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion d'afficheur.

Automate	Port	Caractéristiques
TWDNAC232D	2	Adaptateur de communication équipé d'un port EIA RS-232 à 3 fils avec un connecteur mini DIN. <b>Remarque</b> : Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDNAC485D	2	Adaptateur de communication équipé d'un port EIA RS-485 à 3 fils avec un connecteur mini DIN. <b>Remarque</b> : Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion Afficheur.
TWDNAC485T	2	Adaptateur de communication équipé d'un port EIA RS-485 à 3 fils avec un connecteur de borne. <b>Remarque</b> : Cet adaptateur est disponible uniquement pour les automates 16, 24 et 40 E/S compacts et pour le module d'expansion d'afficheur.
TWDXCPODM	2	Module d'expansion de l'afficheur équipé d'un port EIA RS-232 à 3 fils avec un connecteur mini DIN, d'un port EIA RS-485 à 3 fils avec un connecteur mini DIN et d'un port EIA RS-485 à 3 fils avec un bornier. <b>Remarque</b> : Ce module est disponible uniquement pour les automates modulaires. Lorsque le module est connecté, l'automate ne peut pas disposer d'un module d'expansion de communication.

**Note** : La vérification de la présence du port 2 et de sa configuration (RS232 ou RS485) est uniquement réalisée lors de la mise sous tension ou de la réinitialisation par le microprogramme de l'automate.

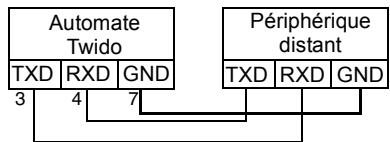
**Câblage nominal** Les connexions de câble nominal sont représentées ci-dessous pour les types EIA RS-232 et EIA RS-485.

**Note :** Si le port 1 est utilisé sur l'automate Twido, le signal DPT sur la broche 5 doit être relié au 0V de la broche 7. Ce signal permet d'indiquer à l'automate Twido que les communications via le port 1 relèvent du protocole Modbus et non du protocole utilisé pour communiquer avec le logiciel TwidoSoft.

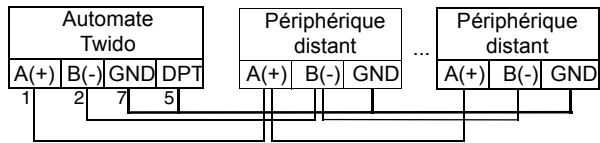
Les connexions de câbles effectuées à chaque périphérique distant sont représentées ci-dessous.

Connexion mini DIN

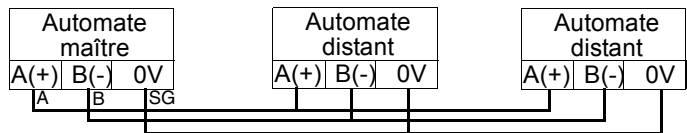
Câble EIA RS-232



Câble EIA RS-485



Connexion bornier



**Configuration logicielle**

Pour configurer l'automate afin d'utiliser une liaison série pour envoyer et recevoir des caractères à l'aide du protocole Modbus, procédez comme suit :

Etape	Description
1	Configurez le port série pour le protocole Modbus à l'aide de TwidoSoft.
2	Créez dans votre application une table d'émission/réception qui sera utilisée par l'instruction EXCHx.

**Configuration du port**

Un automate Twido peut utiliser son port 1 principal ou un port 2 configuré en option pour utiliser le protocole Modbus. Pour configurer un port série pour le protocole Modbus, procédez comme suit :

Etape	Action
1	Définissez tous les modules ou adaptateurs de communication supplémentaires configurés sur la base.
2	Cliquez avec le bouton droit de la souris sur le port, puis cliquez sur Paramétrer les communications de l'automate et modifiez le type du port série sur "Modbus".
3	Définissez les paramètres de communication associés.

**Maître Modbus**

Le mode Modbus maître permet à l'automate d'envoyer une requête Modbus à un esclave et d'attendre la réponse. Le mode Modbus maître n'est pris en charge que par l'intermédiaire de l'instruction EXCHx. Les modes Modbus ASCII et RTU sont tous les deux pris en charge en mode Modbus maître.

La taille maximale des trames émises et/ou reçues s'élève à 250 octets. En outre, la table de mots associée à l'instruction EXCHx se compose des tables de contrôle, d'émission et de réception.

	Octet de poids fort	Octet de poids faible
Table de contrôle	Commande	Longueur (Emission/Réception)
	Décalage réception	Décalage émission
Table d'émission	Octet 1 émis	Octet 2 émis
	...	...
	...	Octet n émis
	Octet n+1 émis	
Table de réception	Octet 1 reçu	Octet 2 reçu
	...	...
	...	Octet p reçu
	Octet p+1 reçu	

**Note :** Outre les requêtes faites à chaque esclave, l'automate maître Modbus peut lancer une requête de **diffusion générale** à tous les esclaves. L'octet **Commande**, dans le cas d'une requête de diffusion générale, doit être réglé sur 00, alors que l'**adresse esclave** doit être réglée sur 0.

<b>Table de contrôle</b>	<p>L'octet <b>Longueur</b> contient la longueur de la table d'émission (250 octets maximum), qui est écrasée par le nombre de caractères reçus à la fin de la réception, si la réception est demandée.</p> <p>Ce paramètre correspond à la longueur en octets de la table d'émission. Si le paramètre de décalage de l'émission est égal à zéro, il sera égal à la longueur de la trame d'émission. Si le paramètre de décalage de l'émission n'est pas égal à zéro, un octet de la table d'émission (indiqué par la valeur de décalage) ne sera pas émis et ce paramètre sera égal à la longueur de la trame plus 1.</p> <p>L'octet <b>Commande</b> doit toujours être égal à 1 (émission et réception) en cas de requête Modbus RTU (sauf pour une diffusion générale).</p> <p>L'octet <b>Décalage émission</b> contient le rang (1 pour le premier octet, 2 pour le deuxième octet, etc.) dans la table d'émission de l'octet à ignorer lors de l'émission des octets. Il est utilisé pour prendre en charge les émissions associées aux valeurs octet/mot dans le cadre du protocole Modbus. Par exemple, si cet octet est égal à 3, le troisième octet est ignoré, ce qui fait du quatrième octet de la table le troisième octet à émettre.</p> <p>L'octet <b>Décalage réception</b> contient le rang (1 pour le premier octet, 2 pour le deuxième octet, etc.) dans la table de réception à ajouter lors de l'émission des octets. Il est utilisé pour prendre en charge les émissions associées aux valeurs octet/mot dans le cadre du protocole Modbus. Par exemple, si cet octet est égal à 3, le troisième octet de la table est renseigné par un ZERO et le troisième octet réellement reçu est entré dans le quatrième emplacement de la table.</p>
--------------------------	---

## Tables d'émission/réception

Dans l'un ou l'autre des modes (Modbus ASCII ou Modbus RTU), la table d'émission est écrite avec le contenu de la requête avant l'exécution de l'instruction EXCHx. Au moment de l'exécution, l'automate détermine quelle est la couche liaison de données et effectue toutes les conversions nécessaires pour traiter l'émission et la réponse. Les caractères de début, de fin et de contrôle ne sont pas stockés dans les tables d'émission/réception.

Une fois que tous les octets ont été émis, l'automate passe en mode de réception et est prêt à recevoir des octets.

La réception se termine de l'une des manières suivantes :

- un délai a été détecté sur un caractère ou une trame,
- le caractère de fin de trame est reçu en mode ASCII,
- la table de réception est saturée.

Les **Octets émis X** contiennent les données (codage RTU) de protocole Modbus à émettre. Si le port de communication est configuré en Modbus ASCII, les caractères de trame corrects sont ajoutés à l'émission. Le premier octet comprend le repère du périphérique (spécifique ou général), le deuxième octet comprend le code de fonction et le reste comprend les informations associées à ce code de fonction.

**Note** : Il s'agit d'une application type, mais toutes les possibilités ne sont pas définies. Aucune validation des données en cours d'émission n'est effectuée.

Les **Octets reçus X** contiennent les données (codage RTU) de protocole Modbus à recevoir. Si le port de communication est configuré en Modbus ASCII, les caractères de trame corrects sont supprimés de la réponse. Le premier octet comprend le repère du périphérique, le deuxième octet comprend le code de fonction (ou code de réponse) et le reste comprend les informations associées à ce code de fonction.

**Note** : Il s'agit d'une application type, mais toutes les possibilités ne sont pas définies. Aucune validation des données en cours de réception n'est effectuée, à l'exception d'une vérification de checksum.

## Esclave Modbus

Le mode Modbus esclave permet à l'automate de répondre à des requêtes Modbus standard provenant d'un maître Modbus.

Lorsque le câble TSXPCX1031 est raccordé à l'automate, la communication avec TwidoSoft démarre sur le port, ce qui désactive temporairement le mode de communication qui était en cours d'exécution avant la connexion de ce câble.

Le protocole Modbus prend en charge deux formats de couche liaison de données : ASCII et RTU. Chaque format est défini par l'implémentation de la couche physique ; le format ASCII utilise sept bits de données tandis que le format RTU en utilise huit.

En mode Modbus ASCII, chaque octet d'un message est envoyé sous la forme de deux caractères ASCII. La trame Modbus ASCII commence par un caractère de début (':') et peut se terminer par deux caractères de fin (CR et LF). Le caractère de fin de trame par défaut est 0x0A (LF). L'utilisateur peut modifier la valeur de cet octet au cours de la configuration. La valeur de contrôle de la trame Modbus ASCII correspond à un simple complément de deux de la trame, excluant les caractères de début et de fin.

Le mode Modbus RTU ne reformate pas le message avant de l'émettre ; cependant, il utilise un mode de calcul de checksum différent, spécifié sous forme de CRC.

Les limitations de la couche liaison de données Modbus sont les suivantes :

- Repère 1-247
- Bits : 128 bits sur demande
- Mots : 125 mots de 16 bits sur demande

**Echange de messages**

Le langage propose deux services pour la communication :

- **Instruction EXCHx** : pour émettre/recevoir des messages.
- **Bloc fonction %MSGx** : pour contrôler les échanges de messages.

L'automate Twido utilise le protocole configuré pour ce port lors du traitement d'une instruction EXCHx.

**Note** : Il est possible de configurer chaque port de communication pour différents protocoles ou pour le même protocole. Pour accéder à l'instruction EXCHx ou au bloc fonction %MSGx de chaque port de communication, il suffit d'ajouter le numéro du port (1 ou 2).

---

**Instruction EXCHx**

L'instruction EXCHx permet à l'automate Twido d'envoyer et/ou de recevoir des informations vers/depuis des périphériques Modbus. L'utilisateur définit une table de mots (%MWi:L) contenant des informations de contrôle, ainsi que les données à envoyer et/ou à recevoir (jusqu'à 250 octets dans l'émission et/ou réception). La description du format de la table de mots a été donnée précédemment.

Un échange de messages s'effectue à l'aide de l'instruction EXCHx.

Syntaxe : [EXCHx %MWi:L]

où : x = numéro du port (1 ou 2)

L = nombre de mots dans les tables de mots de commande, d'émission et de réception

L'automate Twido doit terminer l'échange de la première instruction EXCHx avant de pouvoir en lancer une deuxième. Il est nécessaire d'utiliser le bloc fonction %MSGx lors de l'envoi de plusieurs messages.

Le traitement de l'instruction par liste EXCHx se produit immédiatement, en sachant que toutes les émissions sont démarrées sous contrôle d'interruptions (la réception des données est également sous contrôle d'interruptions), ce qui est considéré comme un traitement en arrière-plan.

---

**Bloc fonction %MSGx**

L'utilisation du bloc fonction %MSGx est facultative ; elle permet de gérer des échanges de données. Le bloc fonction %MSGx remplit trois fonctions :

- **Vérification des erreurs de communications**

La recherche d'erreurs permet de vérifier que le paramètre L (longueur de la table de mots) programmé à l'aide de l'instruction EXCHx est suffisamment grand pour contenir la longueur du message à envoyer. Ce paramètre est comparé à la longueur programmée dans l'octet de poids faible du premier mot de la table de mots.

- **Coordination de plusieurs messages**

Pour garantir la coordination lors de l'envoi de plusieurs messages, le bloc fonction %MSGx fournit les informations requises pour déterminer le moment où l'émission du message précédent est terminée.

- **Emission de messages prioritaires**



Le bloc fonction %MSGx vous permet de suspendre l'émission d'un message afin d'envoyer un message plus urgent.

Le bloc fonction %MSGx dispose d'une entrée et de deux sorties qui lui sont associées :

Entrée/Sortie	Définition	Description
R	Entrée RAZ	A l'état 1 : réinitialise la communication ou le bloc (%MSGx.E = 0 et %MSGx.D = 1).
%MSGx.D	Communication terminée	0 : requête en cours. 1 : communication terminée en cas de fin d'émission, de réception du caractère de fin, d'erreur ou de réinitialisation du bloc.
%MSGx.E	Erreur	0 : longueur du message OK et liaison OK. 1 : en cas de mauvaise commande, de table configurée de manière incorrecte, de mauvais caractère reçu (débit, parité, etc.) ou de saturation de la table de réception.

## Limitations

Il est important de garder à l'esprit les limitations suivantes :

- La présence et la configuration du port 2 (RS232 ou RS485) sont contrôlées lors de la mise sous tension ou de la réinitialisation.
- Tout message en cours de traitement sur le port 1 est abandonné lorsque TwidoSoft est connecté.
- Il est impossible de traiter EXCHx ou %MSG sur un port configuré en tant que liaison distante.
- EXCHx abandonne le traitement Modbus esclave actif.
- Le traitement des instructions EXCHx ne fait pas l'objet d'une nouvelle tentative en cas d'erreur.
- Il est possible d'utiliser l'entrée RAZ pour annuler le traitement de la réception d'une instruction EXCHx.
- Il est possible de configurer des instructions EXCHx avec un délai d'annulation de réception.
- Les messages multiples sont contrôlés via %MSGx.D.

## Erreurs et conditions de fonctionnement

Si une erreur se produit lors de l'utilisation de l'instruction EXCHx, les bits %MSGx.D et %MSGx.E sont réglés sur 1, le mot système %SW63 contient le code d'erreur du port 1 et %SW64 le code d'erreur du port 2.

Mots système	Utilisation
%SW63	Code d'erreur EXCH1 : 0 - opération réussie 1 - nombre d'octets à émettre trop important (> 250) 2 - table d'émission trop petite 3 - table de mots trop petite 4 - débordement de la table de réception 5 - délai écoulé 6 - émission 7 - mauvaise commande dans la table 8 - port sélectionné non configuré/disponible 9 - erreur de réception 10 - impossible d'utiliser %KW en cas de réception 11 - décalage d'émission plus important que la table d'émission 12 - décalage de réception plus important que la table de réception 13 - interruption du traitement EXCH par l'automate
%SW64	Code d'erreur EXCH2 : voir %SW63.

### Redémarrage de l'automate maître

Lorsqu'un automate maître/esclave redémarre, l'un des événements suivants se produit :

- Un démarrage à froid (%S0 = 1) force la réinitialisation des communications.
- Un démarrage à chaud (%S1 = 1) force la réinitialisation des communications.
- En mode Stop, l'automate arrête toutes les communications Modbus.

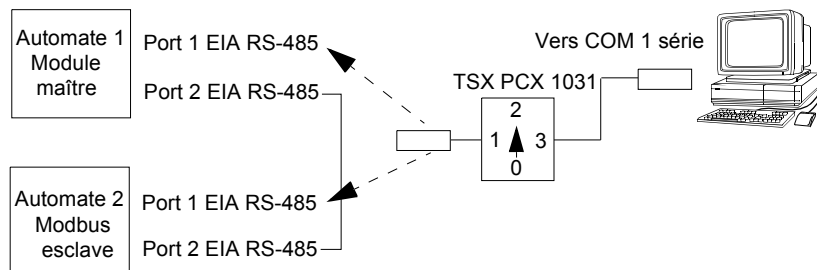
### Exemple 1 de liaison Modbus

Pour configurer une liaison Modbus, procédez comme suit :

1. Configurez le matériel.
2. Connectez le câble de communication Modbus.
3. Configurez le port.
4. Ecrivez une application.
5. Initialisez l'éditeur de tables d'animation.

Les illustrations suivantes représentent l'utilisation de la requête Modbus 3 pour lire des mots de sortie d'un esclave. Cet exemple utilise deux automates Twido.

#### Etape 1 : Configuration du matériel :



La configuration matérielle comprend deux automates Twido. L'un d'entre eux est configuré en tant que Modbus maître et l'autre en tant que Modbus esclave.

**Note :** Dans cet exemple, chaque automate est configuré afin d'utiliser EIA RS-485 sur le port 1 ainsi que EIA RS-485 sur le port 2 optionnel. Sur un automate modulaire, le port 2 optionnel peut être de type TWDNOZ485D ou TWDNOZ485T, ou si vous utilisez TWDXCPODM, il peut être de type TWDNAC485D ou TWDNAC485T. Sur un automate compact, le port 2 optionnel peut être un port TWDNAC485D ou TWDNAC485T.

Pour configurer chaque automate, connectez le câble TSX PCX 1031 au port 1 de l'automate.

**Note :** Le câble TSXPCX1031 peut uniquement être connecté à un automate à la fois et seulement sur le port 1 EIA RS-485.

Connectez ensuite le câble au port COM 1 du PC. Assurez-vous que le commutateur est en position 2. Déchargez et contrôlez l'application. Répétez cette procédure pour le deuxième automate.

**Etape 2 : Connexion du câble de communication Modbus :**

## Connexion mini DIN



## Connexion bornier



Le câblage utilisé dans cet exemple correspond à une simple connexion point à point. Les trois signaux A(+), B(-) et 0V sont câblés conformément à l'illustration. Si vous utilisez le port 1 de l'automate Twido, vous devez connecter le signal DPT (broche 5) au 0V (broche 7). Cette condition du DPT détermine si TwidoSoft est connecté. Lorsqu'il est relié à la terre, l'automate utilise la configuration de port définie dans l'application pour déterminer le type de communication.

**Etape 3 : Configuration du port :**

Matériel -> Ajouter une option TWDNOZ485-	
Matériel => Paramétrage des comm. de l'automate	
Port :	2
Type :	Modbus
Adresse :	1
Débit :	19 200
Données :	8 bits
Parité :	Aucune
Arrêt :	1 bit
Fin de trame :	65
Délai de réponse :	10 x 100 ms
Dépassement trame :	10 ms

Matériel -> Ajouter une option TWDNOZ485-	
Matériel => Paramétrage des comm. de	
Port :	2
Type :	Modbus
Repère :	2
Débit :	19 200
Données :	8 bits
Parité :	Aucune
Arrêt :	1 bit
Fin de trame :	65
Délai de réponse :	100 x 100 ms
Dépassement trame :	10 ms

Dans les applications maître et esclave, les ports EIA RS-485 optionnels sont configurés. Assurez-vous que les paramètres de communication de l'automate sont modifiés en protocole Modbus et à des repères différents.

Dans cet exemple, le maître est réglé sur le repère 1 et l'esclave sur 2. Le nombre de bits est 8, ce qui indique que le mode Modbus RTU sera utilisé. S'il avait été de 7, le mode Modbus ASCII aurait été utilisé. La seule autre valeur par défaut modifiée concerne l'augmentation du délai de réponse à 1 seconde.

**Note :** Etant donné que le mode Modbus RTU a été sélectionné, le paramètre "Fin de trame" a été ignoré.

**Etape 4 :** Ecriture d'une application :

```
LD 1
[%MW0 := 16#0106]
[%MW1 := 16#0300]
[%MW2 := 16#0203]
[%MW3 := 16#0000]
[%MW4 := 16#0004]
LD 1
AND %MSG2.D
[EXCH2 %MW0:11]
LD %MSG2.E
ST %Q0.0
END
```

```
LD 1
[%MW0 := 16#6566]
[%MW1 := 16#6768]
[%MW2 := 16#6970]
[%MW3 := 16#7172]
END
```

A l'aide de TwidoSoft, un programme d'application est écrit pour le maître et l'esclave. Pour l'esclave, il suffit de définir certains mots mémoire sur un ensemble de valeurs connues. Dans le maître, la table de mots de l'instruction EXCHx est initialisée afin de lire quatre mots de l'esclave à l'adresse Modbus 2 qui démarre à l'emplacement %MW0.

**Note :** Remarquez l'utilisation du décalage récepteur défini dans %MW1 du maître Modbus. Le décalage de trois ajoute un octet (valeur = 0) à la troisième position de la zone de réception de la table. Il permet d'aligner les mots dans le maître de façon à ce qu'ils entrent correctement dans les limites de mot. Sans ce décalage, chaque mot de données serait fractionné en deux mots dans le bloc d'échange. Ce décalage est utilisé pour des raisons de commodité.

Avant d'exécuter l'instruction EXCH2, l'application vérifie le bit de communication associé à %MSG2. Finalement, l'état d'erreur du %MSG2 est détecté et stocké sur le premier bit de sortie sur l'E/S de la base automate locale. Il est également possible d'ajouter une recherche d'erreurs supplémentaire à l'aide de %SW64 pour rendre celle-ci plus précise.

**Etape 5 :** Initialisation de l'éditeur de tables d'animation dans le maître :

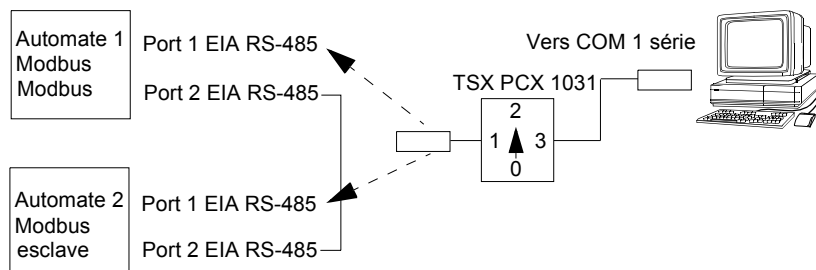
Repère Courant Méorisé Format				
1	%MW5	0203	0000	Hexadécimal
2	%MW6	0008	0000	Hexadécimal
3	%MW7	6566	0000	Hexadécimal
4	%MW8	6768	0000	Hexadécimal
5	%MW9	6970	0000	Hexadécimal
6	%MW10	7172	0000	Hexadécimal

Après le téléchargement et la configuration de tous les automates en vue de leur exécution, ouvrez une table d'animation sur le maître. Examinez la section réponse de la table pour vérifier que le code de réponse correspond à 3 et que le nombre d'octets lus est correct. Notez également, dans cet exemple, que les mots lus de l'esclave (commençant par %MW7) sont correctement alignés avec les limites de mot dans le maître.

## Exemple 2 de liaison Modbus

L'illustration suivante représente l'utilisation de la requête Modbus 16 pour écrire des mots de sortie sur un esclave. Cet exemple utilise deux automates Twido.

**Etape 1 :** Configuration du matériel :



La configuration matérielle est identique à celle de l'exemple précédent.

**Etape 2 :** Connexion du câble de communication Modbus (RS-485) :

Connexion mini DIN



Connexion bornier



Le câblage de communication Modbus est identique à celui de l'exemple précédent.

**Etape 3 : Configuration du port :**

Matériel -> Ajouter une option TWDNOZ485-	Matériel -> Ajouter une option TWDNOZ485-
Matériel => Paramétrage des comm. de l'automate	Matériel => Paramétrage des comm. d
Port : 2	Port : 2
Type : Modbus	Type : Modbus
Adresse : 1	Repère : 2
Débit : 19200	Débit : 19200
Données : 8 bits	Données : 8 bits
Parité : Aucune	Parité : Aucune
Arrêt : 1 bit	Arrêt : 1 bit
Fin de trame : 65	Fin de trame : 65
Délai de réponse : 10 x 100 ms	Délai de réponse : 100 x 100 ms
Dépassement trame : 10 ms	Dépassement trame : 10 ms

Les configurations du port sont identiques à celles de l'exemple précédent.

**Etape 4 : Ecriture d'une application :**

```
LD 1
[%MW0 := 16#010C]
[%MW1 := 16#0007]
[%MW2 := 16#0210]
[%MW3 := 16#0010]
[%MW4 := 16#0002]
[%MW5 := 16#0004]
[%MW6 := 16#6566]
[%MW7 := 16#6768]
LD 1
AND %MSG2.D
[EXCH2 %MW0:11]
LD %MSG2.E
ST %Q0.0
END
```

```
LD 1
[%MW18 := 16#FFFF]
END
```

A l'aide de TwidoSoft, un programme d'application est créé pour le maître et l'esclave. Pour l'esclave, écrivez un seul mot mémoire %MW18. Cette action permet d'allouer de l'espace sur l'esclave pour les adresses mémoire de %MW0 à %MW18. Sans allocation d'espace, la requête Modbus essaierait d'écrire à des emplacements inexistants sur l'esclave.

Dans le maître, la table de mots de l'instruction EXCH2 est initialisée afin d'écrire 4 octets vers l'esclave d'adresse Modbus 2 au repère %MW16 (10 hexadécimal).

**Note :** Remarquez l'utilisation du décalage émission défini dans %MW1 de l'application du maître Modbus. Le décalage de sept permet de supprimer l'octet de poids fort dans le sixième mot (valeur 00 hexadécimale dans %MW5). Cette action permet d'aligner les valeurs de données dans la table d'émission de la table de mots de façon à ce qu'elles entrent correctement dans les limites de mot.

Avant d'exécuter l'instruction EXCH2, l'application vérifie le bit de communication associé à %MSG2. Finalement, l'état d'erreur du %MSG2 est détecté et stocké sur le premier bit de sortie sur l'E/S de la base automate locale. Vous pouvez également effectuer à l'aide de %SW64 une recherche d'erreurs supplémentaire pour rendre celle-ci plus précise.

**Etape 5 :** Initialisation de l'éditeur de tables d'animation :

Création de la table d'animation suivante dans le maître :

Repère	Courant	Mémorisé	Format
1	%MW0	010C	0000 Hexadécimal
2	%MW1	0007	0000 Hexadécimal
3	%MW2	0210	0000 Hexadécimal
4	%MW3	0010	0000 Hexadécimal
5	%MW4	0002	0000 Hexadécimal
6	%MW5	0004	0000 Hexadécimal
7	%MW6	6566	0000 Hexadécimal
8	%MW7	6768	0000 Hexadécimal
9	%MW8	0210	0000 Hexadécimal
10	%MW9	0010	0000 Hexadécimal
11	%MW10	0004	0000 Hexadécimal

Création de la table d'animation suivante dans l'esclave :

Repère	Courant	Mémorisé	Format
1	%MW16	6566	0000 Hexadécimal
2	%MW17	6768	0000 Hexadécimal

Après le déchargement et la configuration de tous les automates en vue de la mise en RUN, ouvrez une table d'animation sur l'automate esclave. Les deux valeurs de %MW16 et %MW17 sont écrites sur l'esclave. Dans le maître, il est possible d'utiliser la table d'animation afin d'examiner la partie table de réception des données d'échange. Ces données affichent le repère de l'esclave, le code de réponse, le premier mot écrit et le nombre de mots écrits à partir de %MW8 dans l'exemple ci-dessus.

---



## Requêtes Modbus standard

### Introduction

Ces requêtes permettent d'échanger des mots ou bits mémoire entre les périphériques. Le format de table utilisé est le même pour le mode RTU et pour le mode ASCII.

Format	Référence
Bit	%Mi
Mot	%MWi

### Maître Modbus : Lecture de N bits

La table suivante représente les requêtes 01 et 02.

	Index de la table	Octet de poids fort	Octet de poids faible
<b>Table de contrôle</b>	0	01 (émission/réception)	06 (Longueur émission) (*)
	1	00 (Décalage réception)	00 (Décalage émission)
<b>Table d'émission</b>	2	Esclave@(1..247)	01 ou 02 (Code de requête)
	3	Numéro du premier bit à lire	
	4	N = Nombre de bits à lire	
<b>Table de réception (après réponse)</b>	5	Esclave@(1..247)	01 (Code de réponse)
	6	Nombre d'octets des données transmis (un octet par bit)	
	7	Premier octet lu (valeur = 00 ou 01)	Deuxième octet lu (si N>1)
	8	Troisième octet lu (si N>1)	
	...		
	(N/2)+6	Nième octet lu (si N>1)	

(\*) Cet octet reçoit également la longueur de la chaîne émise après réponse

**Maître Modbus :** La table suivante représente les requêtes 03 et 04.  
**Lecture de N mots**

	Index de la table	Octet de poids fort	Octet de poids faible
<b>Table de contrôle</b>	0	01 (émission/réception)	06 (Longueur émission) (*)
	1	03 (Décalage réception)	00 (Décalage émission)
<b>Table d'émission</b>	2	Esclave@(1..247)	03 ou 04 (Code de requête)
	3	Numéro du premier mot à lire	
	4	N = Nombre de mots à lire	
<b>Table de réception (après réponse)</b>	5	Esclave@(1..247)	03 (Code de réponse)
	6	00 (octet ajouté à la suite d'une action de Décalage réception)	2*N (nombre d'octets lus)
	7	Premier mot lu	
	8	Deuxième mot lu (si N>1)	
	...		
	N+6	Nième mot lu (si N>2)	

(\*) Cet octet reçoit également la longueur de la chaîne émise après réponse

**Note :** L'opération Décalage réception = 3 ajoute un octet (valeur = 0) à la troisième position de la table de réception. Ce qui assure un bon positionnement dans la table, du nombre d'octets lus et des valeurs des mots lus.

**Maître Modbus :** La table suivante représente la requête 05.  
**Ecriture d'un bit**

	Index de la table	Octet de poids fort	Octet de poids faible
<b>Table de contrôle</b>	0	01 (émission/réception)	06 (Longueur émission) (*)
	1	00 (Décalage réception)	00 (Décalage émission)
<b>Table d'émission</b>	2	Esclave@(1..247)	05 (Code de requête)
	3	Numéro du bit à écrire	
	4	Valeur du bit à écrire	
<b>Table de réception (après réponse)</b>	5	Esclave@(1..247)	05 (Code de réponse)
	6	Numéro du bit écrit	
	7	Valeur écrite	

(\*) Cet octet reçoit également la longueur de la chaîne émise après réponse

**Note :**

- Il n'est pas nécessaire d'utiliser le décalage pour cette requête.
- La trame de la réponse est identique à celle de cette requête (dans un cas normal).
- Pour affecter la valeur 1 à un bit, le mot associé dans la table d'émission doit contenir la valeur FF00H, et 0 pour affecter la valeur 0 à un bit.

**Maître Modbus :** La table suivante représente la requête 06.  
**Ecriture d'un mot**

	Index de la table	Octet de poids fort	Octet de poids faible
<b>Table de contrôle</b>	0	01 (émission/réception)	06 (Longueur émission) (*)
	1	00 (Décalage réception)	00 (Décalage émission)
<b>Table d'émission</b>	2	Esclave@ (1..247)	06 (Code de requête)
	3	Numéro du mot à écrire	
	4	Valeur du mot à écrire	
<b>Table de réception (après réponse)</b>	5	Esclave@ (1..247)	06 (Code de réponse)
	6	Numéro du mot écrit	
	7	Valeur écrite	

(\*) Cet octet reçoit également la longueur de la chaîne émise après réponse

**Note :**

- Il n'est pas nécessaire d'utiliser le décalage pour cette requête.
- La trame de la réponse est identique à celle de cette requête (dans un cas normal).

**Maître Modbus :** La table suivante représente la requête 15.  
**Ecriture de N bits**

	Index de la table	Octet de poids fort	Octet de poids faible
<b>Table de contrôle</b>	0	01 (émission/réception)	8 + nombre d'octets (émission)
	1	00 (Décalage réception)	07 (Décalage émission)
<b>Table d'émission</b>	2	Esclave@(1..247)	15 (Code de requête)
	3	Numéro du premier bit à écrire	
	4	$N_1$ = Nombre de bits à écrire	
	5	00 (octet non envoyé, effet de décalage)	$N_2$ = Nombre d'octets des données à écrire
	6	Valeur du premier octet	Valeur du second octet
<b>Table de contrôle</b>	7	Valeur du troisième octet	Valeur du quatrième octet
	...		
<b>Table d'émission</b>	$6+(N_2/2)$	Valeur du $N_2$ ième octet	
<b>Table de réception (après réponse)</b>		Esclave@(1..247)	15 (Code de réponse)
		Numéro du premier bit écrit	
		Nombre de bits écrits (= $N_1$ )	

**Note :**

- L'opération Décalage émission = 7 supprime le 7ième octet de la trame envoyée. Elle permet également d'assurer une bonne correspondance entre les valeurs des mots de la table d'émission.

**Maître Modbus :** La table suivante représente la requête 16.

**Ecriture de N mots**

	Index de la table	Octet de poids fort	Octet de poids faible
<b>Table de contrôle</b>	0	01 (émission/réception)	8 + (2*N) (Longueur émission)
	1	00 (Décalage réception)	07 (Décalage émission)
<b>Table d'émission</b>	2	Esclave@(1..247)	16 (Code de requête)
	3	Numéro du premier mot à écrire	
	4	N = Nombre de mots à écrire	
	5	00 (octet non envoyé, effet de décalage)	2*N = Nb d'octets à écrire
	6	Première valeur du mot à écrire	
	7	Deuxième valeur à écrire	
	...		
	N+5	N valeurs à écrire	
<b>Table de réception (après réponse)</b>	N+6	Esclave@(1..247)	16 (Code de réponse)
	N+7	Numéro du premier mot écrit	
	N+8	Nombre de mots écrits (= N)	

**Note :** L'opération Décalage émission = 7 supprime le 5ème octet MMSB de la trame envoyée. Elle permet également d'assurer une bonne correspondance entre les valeurs des mots de la table d'émission.

---

## Vue d'ensemble des communications TCP/IP Ethernet

---

### Fonctionnalités Ethernet

Ce sous-chapitre décrit les fonctionnalités de la base automate Twido TWDLCAE40DRF prenant en charge Ethernet. La base automate TWDLCAE40DRF prend en charge Ethernet et implémente le protocole d'application Modbus (MBAP) sur TCP/IP. Le protocole Modbus TCP/IP autorise les communications poste à poste via le réseau dans une topologie client/serveur.

### Format de trame

L'automate compact Twido TWDLCAE40DRF prend uniquement en charge le format de trame Ethernet II. Il ne prend pas en charge la trame IEEE802.3. Remarque : D'autres automates disponibles auprès de Schneider Electric, par exemple les gammes Premium et Quantum, prennent en charge à la fois les formats de trame Ethernet II et IEEE802.3. Ils peuvent également être sélectionnés selon leur format de trame. Par conséquent, si vous souhaitez associer votre automate Twido avec des automates Premium ou Quantum, vous devrez les configurer pour l'utilisation du format de trame Ethernet II afin d'obtenir une compatibilité optimale.

### Connexions TCP

L'automate compact TWDLCAE40DRF est un dispositif 4 voies simultanées prenant en charge la communication sur un réseau Ethernet 100Base-TX. Il implémente l'autonégociation 100Base-TX et peut également fonctionner sur un réseau 10Base-T. De plus, il permet une connexion IP repérée, telle que configurée dans le programme d'application TwidoSoft (pour plus d'informations sur le format IP repérée, voir *Onglet IP repérée*, p. 177).

### Adresse IP

Une adresse IP statique est affectée en tant qu'adresse par défaut à chaque base automate TWDLCAE40DRF. L'adresse IP par défaut du périphérique est dérivée de l'adresse physique MAC unique (adresse internationale IEEE) qui est stockée en permanence dans l'automate compact. Pour augmenter la flexibilité de votre réseau sans recourir à l'adresse IP par défaut, le programme TwidoSoft vous permet de configurer une adresse IP statique différente pour ce périphérique et de définir les adresses IP du sous-réseau et de la passerelle.

### Modbus TCP Client/Serveur

Un automate TWDLCAE40DRF peut être à la fois Client et Serveur Modbus TCP/IP selon qu'il interroge ou répond à un périphérique distant. Le service de messagerie TCP est implémenté via le port TCP 502. Le client Modbus est implémenté via l'instruction %EXCH3 et la fonction %MSG3. Vous pouvez programmer plusieurs instructions %EXCH3, mais une seule peut être active au même moment. La connexion TCP est automatiquement négociée par l'automate compact dès que l'instruction %EXCH3 est activée.

## Guide de configuration rapide TCP/IP pour les communications Ethernet PC vers automate

### Champ d'application

Ce guide de configuration rapide TCP/IP fournit des informations sur la connexion Ethernet et la configuration TCP/IP, et facilite ainsi la configuration des communications entre le PC exécutant l'application TwidoSoft et l'automate Twido sur un réseau Ethernet autonome.

### Vérification des paramètres IP en cours du PC

La procédure suivante indique comment vérifier les paramètres IP en cours de votre PC. Elle s'applique à toutes les versions du système d'exploitation Windows.

Etape	Action
1	Sélectionnez l'option <b>Exécuter</b> du menu <b>Démarrer</b> de Windows.
2	Saisissez " <b>command</b> " dans la zone de texte <b>Ouvrir</b> de la boîte de dialogue Exécuter. <b>Résultat</b> : L'invite <b>C:\WINDOWS\system32\command.com</b> apparaît.
3	Saisissez " <b>ipconfig</b> " à l'invite.
4	La fenêtre <b>Configuration IP</b> de Windows apparaît et contient les paramètres suivants : Adresse IP : Masque de sous-réseau : Passerelle par défaut : <b>Remarque</b> : Les paramètres IP ci-dessus ne peuvent être modifiés directement à l'invite. Ils sont disponibles uniquement pour consultation. Pour modifier la configuration IP de votre PC, reportez-vous à la section suivante.



## Configuration des paramètres TCP/IP du PC

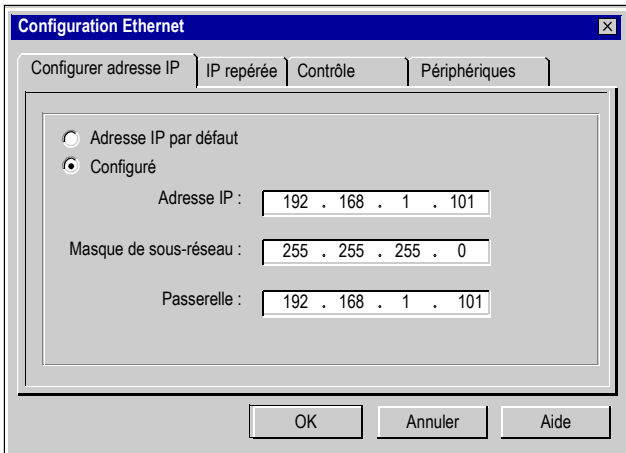
Les informations suivantes expliquent comment configurer les paramètres TCP/IP de votre PC exécutant l'application TwidoSoft pour la programmation et le contrôle de l'automate Twido sur le réseau. La procédure décrite ci-dessous s'applique à un PC équipé du système d'exploitation Windows XP, et est donnée à titre d'exemple uniquement. (Pour les autres systèmes d'exploitation, reportez-vous aux instructions de configuration TCP/IP présentes dans le guide utilisateur du système d'exploitation installé sur votre PC.)

Etape	Action
<b>Remarque : Si votre PC est déjà installé et que la carte Ethernet est configurée sur le réseau autonome existant, vous n'avez pas besoin de modifier les paramètres de l'adresse IP (ignorez les étapes 1 à 6 et passez à la section suivante). Suivez les étapes 1 à 6 si vous voulez modifier les paramètres TCP/IP de votre PC.</b>	
1	Sélectionnez l'option <b>Panneau de configuration &gt; Connexions réseau</b> du menu <b>Démarrer</b> de Windows.
2	Cliquez avec le bouton droit sur l'icône <b>Connexion au réseau local</b> (le réseau autonome) sur laquelle vous voulez installer l'automate Twido, puis cliquez sur <b>Propriétés</b> .
3	Sélectionnez <b>TCP/IP</b> dans la liste des composants de réseau installés, puis cliquez sur <b>Propriétés</b> . <b>Remarque :</b> Si le protocole TCP/IP ne figure pas dans la liste des composants installés, reportez-vous au guide utilisateur de votre système d'exploitation pour installer le composant de réseau TCP/IP.
4	La boîte de dialogue <b>Propriétés de Protocole Internet (TCP/IP)</b> apparaît. Elle contient les paramètres TCP/IP actuels de votre PC, y compris l' <b>adresse IP</b> et le <b>masque de sous-réseau</b> . <b>Remarque :</b> Sur un réseau autonome, n'utilisez pas l'option <b>Obtenir une adresse IP automatiquement</b> . La case d'option <b>Spécifier une adresse IP</b> doit être sélectionnée et les champs Adresse IP et Masque de sous-réseau doivent être renseignés avec des paramètres IP valides.
5	Saisissez une <b>adresse IP statique</b> valide en notation décimale séparée par des points. Sur un réseau autonome, nous vous recommandons de spécifier une adresse IP réseau de classe C (voir <i>Adressage IP</i> , p. 167). Par exemple, 192.168.1.198 est une adresse IP de classe C. <b>Remarque :</b> L'adresse IP spécifiée doit être compatible avec l'ID de réseau du réseau existant. Par exemple, si le réseau existant prend en charge les adresses IP de type 192.168.1.xxx (où 192.168.1 est l'ID de réseau et xxx = 0-255 est l'ID d'hôte), vous pouvez spécifier 191.168.1.198 comme adresse IP valide sur votre PC. (Assurez-vous que l'ID d'hôte 198 est unique sur le réseau.)

Etape	Action
<b>Remarque : Si votre PC est déjà installé et que la carte Ethernet est configurée sur le réseau autonome existant, vous n'avez pas besoin de modifier les paramètres de l'adresse IP (ignorez les étapes 1 à 6 et passez à la section suivante). Suivez les étapes 1 à 6 si vous voulez modifier les paramètres TCP/IP de votre PC.</b>	
6	Saisissez un <b>masque de sous-réseau</b> valide en notation décimale séparée par des points. Si votre réseau de classe C ne comporte pas de sous-réseaux, nous vous recommandons de spécifier un masque de sous-réseau de réseau de classe C par défaut tel que 255 . 255 . 255 . 0.

### Configuration des paramètres TCP/IP de l'automate Twido

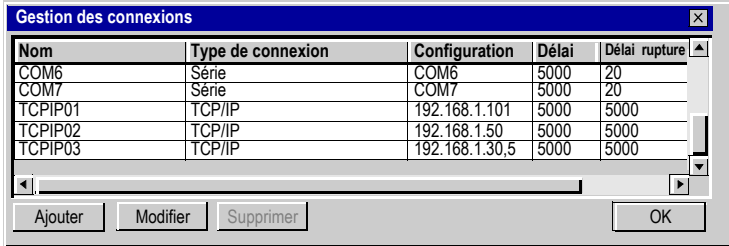
Après avoir configuré les paramètres TCP/IP du PC exécutant l'application TwidoSoft, vous devez configurer les paramètres TCP/IP de l'automate Twido qui communiquera sur le réseau via cette application. Pour ce faire, procédez comme suit :

Etape	Action
1	Connectez le PC exécutant TwidoSoft au port console RS-485 de l'automate Twido à l'aide d'un câble série (TSXPCX 1031).
2	Lancez le programme d'application <b>TwidoSoft</b> sur votre PC.
3	Sélectionnez un nouveau <b>matériel</b> dans le Navigateur application TwidoSoft et choisissez l'automate <b>TWDLCAE40DRF</b> .
4	Sélectionnez <b>Automate &gt; Sélectionner une connexion</b> dans la barre de menus TwidoSoft, puis choisissez le port <b>COM1</b> .
5	<p>Cliquez deux fois sur l'icône <b>Port Ethernet</b> dans le Navigateur application TwidoSoft (ou sélectionnez <b>Matériel &gt; Ethernet</b> dans la barre de menus) pour afficher la boîte de dialogue <b>Configuration Ethernet</b>. Celle-ci se présente de la manière suivante :</p> 

Etape	Action
6	<p>Dans l'onglet <b>Configurer adresse IP</b>, sélectionnez la case d'option <b>Configuré</b> et configurez les adresses IP, de masque de sous-réseau et de passerelle (cf. étapes 7 à 9).</p> <p><b>Remarque</b> : A ce stade, nous traitons uniquement la configuration de base d'une communication entre un PC et un automate sur le réseau Ethernet. Par conséquent, nous n'allons pas configurer les onglets IP repérée, Contrôle d'inactivité et Périphériques distants pour l'instant.</p>
7	<p>Saisissez une <b>adresse IP</b> statique valide pour l'automate Twido en notation décimale séparée par des points. Cette adresse IP doit être compatible avec celle du PC que vous avez configurée dans la section précédente.</p> <p><b>Remarque</b> : Les adresses IP de l'automate Twido et du PC doivent partager le même ID de réseau. Cependant, l'ID d'hôte de l'automate Twido doit être unique sur le réseau et différent de celui du PC. Par exemple, si l'adresse IP de classe C du PC est 192.168.1.198, l'adresse valide de l'automate Twido est alors 192.168.1.xxx (où 192.168.1 est l'ID de réseau et xxx = 0-197, 199-255 est l'ID d'hôte).</p>
8	<p>Saisissez un <b>masque de sous-réseau</b> valide en notation décimale séparée par des points. L'automate Twido et le PC exécutant TwidoSoft doivent se trouver sur le même segment de réseau. Par conséquent, vous devez saisir un masque de sous-réseau identique à celui spécifié pour le PC.</p> <p><b>Remarque</b> : Si le masque de sous-réseau n'est pas utilisé dans votre réseau de classe C, nous vous recommandons de spécifier un masque de sous-réseau de réseau de classe C par défaut tel que 255.255.255.0.</p>
9	<p>Saisissez une adresse de <b>passerelle</b> valide en notation décimale séparée par des points.</p> <p><b>Remarque</b> : Si votre réseau autonome ne dispose pas de passerelle, saisissez dans ce champ l'adresse IP de l'automate Twido que vous avez configurée à l'étape 6.</p>
10	<p>Cliquez sur <b>OK</b> pour enregistrer les paramètres de configuration Ethernet de l'automate Twido.</p>

**Configuration  
d'une nouvelle  
connexion TCP/  
IP dans  
TwidoSoft**

Vous allez maintenant configurer une nouvelle connexion TCP/IP dans l'application TwidoSoft. La nouvelle connexion TCP/IP dédiée permettra au PC exécutant TwidoSoft et à l'automate Twido de communiquer sur le réseau Ethernet.

Etape	Action
1	<p>Sélectionnez <b>Fichier &gt; Préférences &gt; Gestion des connexions</b> dans la barre de menus TwidoSoft pour afficher la boîte de dialogue <b>Gestion des connexions</b>. Celle-ci se présente de la manière suivante :</p> 
2	<p>Cliquez sur le bouton <b>Ajouter</b> dans la boîte de dialogue <b>Gestion des connexions</b>.</p> <p><b>Résultat</b> : Une ligne de connexion supplémentaire est ajoutée. Elle comprend les paramètres de connexion par défaut conseillés. Vous devez modifier ces paramètres.</p> <p><b>Remarque</b> : Deux méthodes vous sont proposées pour modifier la valeur d'un champ :</p> <ul style="list-style-type: none"><li>● Cliquez dans le champ voulu, puis sur le bouton <b>Modifier</b>.</li><li>● Cliquez deux fois dans le champ voulu.</li></ul>
3	<p>Dans le champ <b>Nom</b>, saisissez un nom descriptif pour la nouvelle connexion. Un nom valide contient au maximum 32 caractères alphanumériques.</p>
4	<p>Cliquez dans le champ <b>Type de connexion</b> pour dérouler la liste et sélectionnez <b>TCP/IP</b> puisque vous configurez une nouvelle connexion Ethernet entre un PC et un automate Twido prenant en charge Ethernet.</p>
5	<p>Dans le champ <b>Configuration</b>, saisissez une <b>adresse IP</b> et un <b>ID d'unité</b> valides (le cas échéant) qui correspondent aux informations IP de l'automate Twido TWDLCAE40DRF auquel vous souhaitez vous connecter. L'adresse IP et l'ID d'unité doivent être séparés par une virgule.</p> <p><b>Adresse IP</b> : Saisissez l'adresse IP statique de votre automate Twido spécifiée dans la section précédente.</p> <p><b>ID unité</b> : Ne renseignez pas cette partie du champ sauf si vous vous connectez à un automate Twido situé sur une liaison série Modbus via un pont.</p>

---

Etape	Action
6	Utilisez les paramètres par défaut pour les champs <b>Délai</b> et <b>Délai avant rupture</b> , à moins que vous n'ayez des besoins spécifiques. (Pour plus d'informations, voir <i>Gestion des connexions Ethernet</i> , p. 185.)
7	Cliquez sur <b>OK</b> pour enregistrer les nouveaux paramètres de connexion et fermer la boîte de dialogue Gestion des connexions. <b>Résultat</b> : Les noms de toutes les nouvelles connexions sont ajoutés à la liste déroulante des connexions dans la boîte de dialogue <b>Fichier &gt; Préférences</b> et dans <b>Automate &gt; Sélectionner une connexion</b> .

---

## Connexion de l'automate au réseau

### Présentation

Les informations suivantes décrivent l'installation de votre automate compact TDWLCAE40DRF sur votre réseau Ethernet.

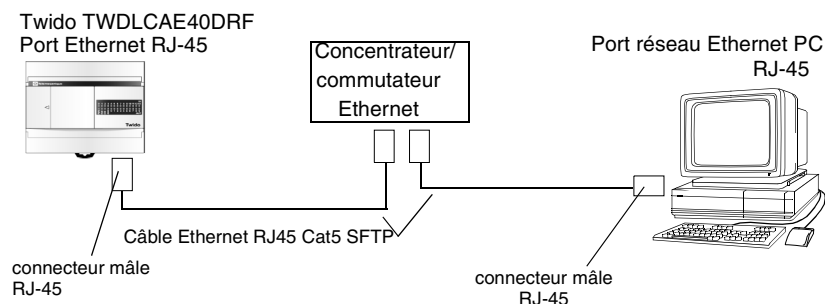
### Détermination du groupe d'adresses IP approprié

Contactez votre administrateur réseau pour déterminer si vous devez configurer un nouvel ensemble d'adresses IP, d'adresses de passerelle et de masque de sous-réseau pour vos équipements. Si l'administrateur affecte de nouveaux paramètres d'adresse IP, vous devez saisir ces informations manuellement dans l'application TwidoSoft. Suivez les instructions de la section *Configuration TCP/IP*, p. 173 ci-dessous.

### Connexion par réseau Ethernet

**Note :** Même si la connexion directe par câble (à l'aide d'un câble inverseur ) est prise en charge entre l'automate Twido TDWLCAE40DRF et le PC exécutant le logiciel de programmation TwidoSoft, nous déconseillons cette méthode. Par conséquent, préférez toujours une connexion via un concentrateur/commutateur réseau Ethernet.

L'illustration suivante représente une connexion réseau Twido via un concentrateur/commutateur Ethernet :



L'automate Twido TDWLCAE40DRF dispose d'un connecteur RJ-45 pour la liaison au réseau Ethernet 100 BASE-TX prenant en charge l'autonégociation. Il prend en charge les vitesses de connexion réseau de 100 Mbit/s et 10 Mbit/s.

**Note :** Utilisez toujours un câble Ethernet catégorie 5 pour connecter l'automate Twido à un réseau 100Base-TX.

---

## Adressage IP

---

### Présentation

Ce sous-chapitre fournit des informations relatives à la notation des adresses IP et aux concepts de sous-réseau et de passerelle.

---

### Adresse IP

Une adresse IP est une quantité 32 bits exprimée en notation décimale séparée par des points. Elle consiste en quatre groupes de nombres dont la valeur est comprise entre 0 et 255 et qui sont séparés les uns des autres par un point. Par exemple, 192.168.2.168 est une adresse IP en notation décimale séparée par des points (remarquez que cette adresse IP réservée est donnée à titre d'exemple uniquement).

Sur les réseaux habituels, les adresses IP sont regroupées en trois catégories : les réseaux de classe A, B et C. Les classes se différencient selon la valeur de leur premier numéro (cf. tableau ci-dessous).

Premier numéro	Classe IP
0-127	Classe A
128-191	Classe B
192-223	Classe C

---

### Masque de sous-réseau IP

Une adresse IP est constituée de deux parties, l'ID de réseau et l'ID d'hôte. Le masque de sous-réseau est utilisé pour séparer la partie réseau de l'adresse IP afin de créer artificiellement des sous-réseaux avec des ID d'hôte plus nombreux. Ainsi, le sous-réseau permet de connecter plusieurs réseaux physiques à des réseaux logiques. Tous les périphériques d'un même sous-réseau partagent le même ID de réseau.

Tous les périphériques du même sous-réseau partagent le même ID de réseau.

**Note :** Si vous faites partie d'une grande société, il est très probable que les réseaux de votre entreprise utilisent des sous-réseaux. Lors de l'installation de votre nouvel automate Twido sur le réseau existant, consultez votre administrateur réseau pour obtenir des informations sur les sous-réseaux.

---

**Adresse de passerelle**

La passerelle est un périphérique de sous-réseau (également appelé routeur) qui permet à votre segment réseau d'accéder à d'autres segments réseau du réseau global de votre entreprise, à Internet ou à un Intranet distant. L'adresse de passerelle utilise le même format en notation décimale séparée par des points que celui de l'adresse IP décrit ci-dessus.

**Note :** Lors de l'installation de votre nouvel automate Twido sur le réseau existant, consultez votre administrateur réseau pour obtenir des informations sur les passerelles.



---

## Affectation d'adresses IP

---

### Présentation

Ce sous-chapitre fournit des informations concernant la détermination du type d'adresse IP à affecter à l'automate Twido TWDLCAE40DRF que vous voulez installer sur le réseau.

### Installation sur un réseau autonome

L'automate Twido TWDLCAE40DRF est conçu pour être installé sur un réseau Ethernet autonome.

**Note :** Un réseau est autonome lorsqu'il n'est pas relié à Internet ou au réseau Intranet d'une entreprise.

### Adresse MAC et adresse IP par défaut de l'automate

**Adresse MAC :** Chaque automate Twido TWDLCAE40DRF dispose de sa propre adresse MAC définie en usine. Il s'agit d'une adresse mondiale unique de 48 bits affectée à chaque périphérique Ethernet.

**Adresse IP par défaut :** L'adresse IP par défaut de l'interface Ethernet de l'automate Twido est dérivée de son adresse MAC unique.

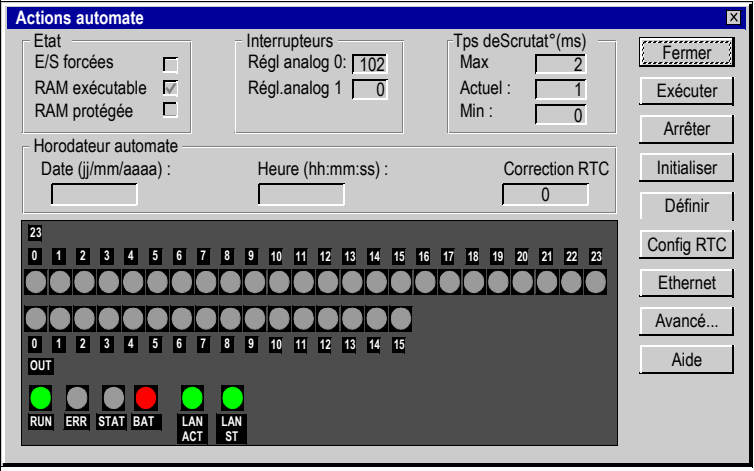
L'adresse IP par défaut, exprimée en notation décimale séparée par des points, se définit comme suit :

085.016.xxx.yyy, où :

- 085.016. est un en-tête défini partagé par toutes les adresses IP dérivées d'une adresse MAC,
  - xxx et yyy sont les deux derniers nombres de l'adresse MAC du périphérique. Par exemple, l'adresse IP dérivée de l'adresse MAC 00.80.F4.81.01.11 est 085.016.001.11.
-

Vérification de l'adresse MAC et de l'adresse IP actuelle de l'automate

Pour vérifier l'adresse MAC et l'adresse IP actuelle de l'automate Twido, ainsi que les paramètres de configuration IP (adresses de masque de sous-réseau et de passerelle) et l'état de la connexion Ethernet, procédez comme suit :

Etape	Action
1	Sélectionnez <b>Automate</b> dans la barre de menus du programme d'application TwidoSoft.
2	<div>Sélectionnez <b>Vérifier l'automate</b> dans la liste des éléments de menu. <b>Résultat</b> : La boîte de dialogue <b>Actions automate</b> apparaît. Elle affiche les voyants Twido sur une face avant et se présente de la manière suivante :</div> <div></div>

Etape	Action																												
3	<p>Cliquez sur le bouton <b>Ethernet</b> situé dans la partie droite de l'écran pour accéder aux paramètres de connexion.</p> <p><b>Résultat</b> : Le tableau <b>Actions automate - Ethernet</b> apparaît. Il contient les informations concernant l'adresse MAC, l'adresse IP actuelle, le sous-réseau et la passerelle, ainsi que des informations sur la connexion Ethernet. Le tableau se présente de la manière suivante :</p> <div><div>Actions automate - Ethernet</div><table><tr><td>Adresse MAC Ethernet</td><td>00 80 f4 10 00 3a</td></tr><tr><td>Adresse IP</td><td>192.168.2.168</td></tr><tr><td>Passerelle par défaut</td><td></td></tr><tr><td>Masque de sous-réseau</td><td>255.255.255.0</td></tr><tr><td>Etat V1</td><td>Serveur passif, utilisé par P-Unit (192.168.2.2)</td></tr><tr><td>Etat V2</td><td>Serveur au repos</td></tr><tr><td>Etat V3</td><td>Serveur au repos</td></tr><tr><td>Etat V4</td><td>Serveur au repos</td></tr><tr><td>Paquets reçus</td><td>198</td></tr><tr><td>Paquets envoyés</td><td>197</td></tr><tr><td>Erreurs de paquets reçus</td><td>0</td></tr><tr><td>Paquets Sans réponse</td><td>0</td></tr><tr><td>STAT Ethernet</td><td>Fonctionnement normal</td></tr><tr><td>Vit. de conect. courante</td><td>100M</td></tr></table><div><div>Fermer</div><div>Aide</div><div>Effacer</div></div></div>	Adresse MAC Ethernet	00 80 f4 10 00 3a	Adresse IP	192.168.2.168	Passerelle par défaut		Masque de sous-réseau	255.255.255.0	Etat V1	Serveur passif, utilisé par P-Unit (192.168.2.2)	Etat V2	Serveur au repos	Etat V3	Serveur au repos	Etat V4	Serveur au repos	Paquets reçus	198	Paquets envoyés	197	Erreurs de paquets reçus	0	Paquets Sans réponse	0	STAT Ethernet	Fonctionnement normal	Vit. de conect. courante	100M
Adresse MAC Ethernet	00 80 f4 10 00 3a																												
Adresse IP	192.168.2.168																												
Passerelle par défaut																													
Masque de sous-réseau	255.255.255.0																												
Etat V1	Serveur passif, utilisé par P-Unit (192.168.2.2)																												
Etat V2	Serveur au repos																												
Etat V3	Serveur au repos																												
Etat V4	Serveur au repos																												
Paquets reçus	198																												
Paquets envoyés	197																												
Erreurs de paquets reçus	0																												
Paquets Sans réponse	0																												
STAT Ethernet	Fonctionnement normal																												
Vit. de conect. courante	100M																												
4	<p>Remarque : L'adresse <b>MAC</b> unique de l'automate Twido apparaît dans la première ligne du tableau.</p>																												
5	<p>Les informations IP affichées dans ce tableau varient en fonction des paramètres définis par l'utilisateur dans l'onglet <b>Configurer IP</b> de la boîte de dialogue <b>Configuration Ethernet</b> (voir l'<i>Onglet Configurer adresse IP</i>, p. 175) :</p> <ul style="list-style-type: none"><li>• Si vous avez sélectionné <b>Adresse IP par défaut</b> dans l'onglet Configurer adresse IP, le tableau ci-dessus affichera l'adresse IP par défaut (dérivée de l'adresse MAC) de l'automate Twido, ainsi que le sous-réseau et la passerelle par défaut.</li><li>• Si vous avez sélectionné <b>Configuré</b> dans l'onglet Configurer adresse IP, le tableau ci-dessus affichera les paramètres de l'adresse IP actuelle, du sous-réseau et de la passerelle définis précédemment dans l'onglet Configurer adresse IP.</li></ul> <p>Remarque : Les champs restants fournissent des informations sur l'état actuel de la connexion Ethernet. Pour plus d'informations, reportez-vous au chapitre (Voir TwdoSOFT).</p>																												

**Adresses IP  
privées**

Si votre réseau est autonome (non relié à Internet), vous pouvez affecter une adresse IP à votre nœud de réseau (automate Twido) de manière arbitraire (tant que l'adresse IP est conforme à la règle de notation de l'IANA et qu'elle n'entre pas en conflit avec l'adresse IP d'un autre périphérique connecté au réseau).

Les adresses IP privées satisfont aux besoins d'adressage IP arbitraire sur un réseau autonome. Remarque : Les adresses situées dans cet espace d'adresses privées ne sont uniques qu'au sein de l'entreprise.

Le tableau suivant présente l'espace réservé aux adresses IP privées :

Réseau	Plage valide d'adresses IP privées
Classe A	10.0.0.0 -> 10.255.255.255
Classe B	172.16.0.0 -> 172.31.255.255
Classe C	192.168.0.0 -> 192.168.255.255

**Affectation d'une  
adresse IP à  
l'automate**

Les réseaux actuels sont rarement totalement isolés par rapport à Internet ou au reste du réseau Ethernet d'une entreprise. Par conséquent, si vous installez et connectez votre base automate Twido sur un réseau existant, n'affectez pas d'adresse IP de manière arbitraire sans consulter auparavant votre administrateur réseau. Vous devrez suivre les instructions décrites ci-après lorsque vous affecterez une adresse IP à l'automate.

<b>Note</b> : Il est recommandé d'utiliser les adresses IP de classe C sur les réseaux autonomes.
---

## Configuration TCP/IP

---

### Présentation

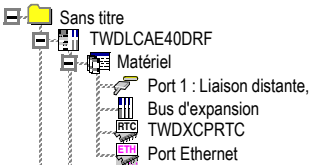
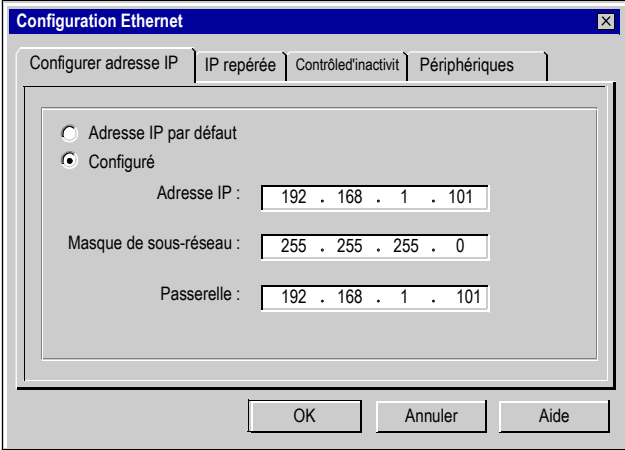
Ce sous-chapitre fournit les instructions détaillées de configuration TCP/IP Ethernet de votre automate compact Twido TWDLCAE40DRF.

<p><b>Note</b> : La configuration TCP/IP peut être effectuée uniquement lorsque le programme d'application TwidoSoft est en mode local.</p>
---

---

Appel de la boîte de dialogue Configuration Ethernet

Les étapes suivantes décrivent en détail l'appel de la boîte de dialogue **Configuration Ethernet**.

Etape	Action
1	<p>Ouvrez le <b>Navigateur application</b>, comme le montre l'illustration suivante.</p> <p><b>Résultat :</b></p>  <p><b>Remarque :</b> Assurez-vous qu'un périphérique prenant en charge Ethernet, TWDLCAE40DRF par exemple, est sélectionné en tant que matériel courant afin que l'option matérielle Port Ethernet apparaisse.</p>
2	<p>Cliquez deux fois sur l'icône <b>Port Ethernet</b> afin d'ouvrir la boîte de dialogue <b>Configuration Ethernet</b> (cf. illustration ci-dessous).</p> <p><b>Résultat :</b></p>  <p><b>Remarque :</b> Deux autres méthodes permettent d'afficher l'écran Configuration Ethernet :</p> <ol style="list-style-type: none"><li>1. Cliquez avec le bouton droit de la souris sur l'icône <b>Port Ethernet</b> et sélectionnez <b>Editer</b> de la liste contextuelle.</li><li>2. Sélectionnez <b>Matériel &gt; Ethernet</b> dans la barre de menus TwidoSoft.</li></ol>

Configuration TCP/IP

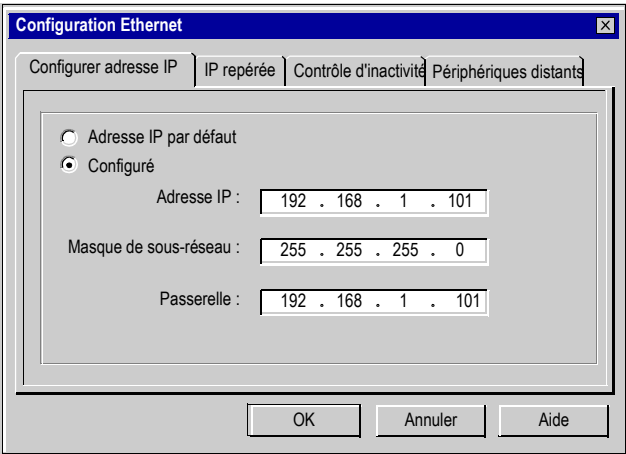
Les sections suivantes décrivent de façon détaillée la configuration des paramètres TCP/IP de Twido TWDLCAE40DRF à l'aide des onglets **Configurer adresse IP**, **IP repérée**, **Contrôle d'inactivité** et **Périphériques distants**.

## Onglet Configurer adresse IP

**Présentation** Ce sous-chapitre explique comment configurer l'onglet Configurer adresse IP de la boîte de dialogue Configuration Ethernet.

**Note :** L'adresse IP de l'automate Twido peut être configurée uniquement lorsque le programme d'application TwidoSoft est en mode local.

**Onglet Configurer adresse IP** L'illustration suivante présente une capture de l'onglet Configurer adresse IP, contenant des exemples d'adresses IP, de passerelle et de sous-réseau configurées manuellement par l'utilisateur :



**Configuration de l'onglet Configurer adresse IP** Les informations suivantes expliquent comment configurer les différents champs de l'onglet Configurer adresse IP :

Champ	Configuration
Adresse IP par défaut	Sélectionnez cette case d'option si vous ne voulez pas définir l'adresse IP de votre automate Twido manuellement (les zones de texte Adresse IP, Masque de sous-réseau et Passerelle sont grisées). L'automate Twido utilise alors l'adresse IP par défaut de l'interface Ethernet dérivée de son adresse MAC. <b>Remarque :</b> Pour plus d'informations sur l'adresse MAC, reportez-vous à la rubrique <i>Affectation d'adresses IP</i> , p. 169.
Configuré	Sélectionnez cette case d'option pour configurer manuellement les adresses IP, de sous-réseau et de passerelle. <b>Remarque :</b> Consultez votre administrateur réseau ou système pour obtenir les paramètres IP valides du réseau.

Champ	Configuration
Adresse IP	<p>Saisissez l'adresse IP statique de votre automate en notation décimale séparée par des points.</p> <p><b>Attention :</b> Pour une bonne communication entre les périphériques, les adresses IP du PC exécutant l'application TwidoSoft et de l'automate doivent partager le même ID de réseau.</p> <p><b>Remarque :</b> Pour une bonne communication sur le réseau, les périphériques connectés doivent avoir une adresse IP unique. Lorsque l'automate Twido est connecté au réseau, il recherche la présence d'adresses IP doubles. Si une adresse IP double est détectée sur le réseau, le voyant LAN ST de l'automate clignote 4 fois à intervalles réguliers. Vous devez alors renseigner ce champ avec une adresse IP unique.</p>
Masque de sous-réseau	<p>Saisissez le masque de sous-réseau valide affecté à l'automate par votre administrateur réseau. Notez que ce champ ne peut rester vierge ; vous devez le renseigner.</p> <p>Par défaut, l'application TwidoSoft calcule automatiquement et affiche le masque de sous-réseau par défaut en fonction de la classe de l'adresse IP définie dans le champ précédent. Selon la catégorie d'adresse IP réseau de l'automate, les valeurs du masque de sous-réseau par défaut respectent la règle suivante :</p> <p>Réseau de classe A -&gt; Masque de sous-réseau par défaut : 255.0.0.0 Réseau de classe B -&gt; Masque de sous-réseau par défaut : 255.255.0.0 Réseau de classe C -&gt; Masque de sous-réseau par défaut : 255.255.255.0</p> <p><b>Attention :</b> Pour une bonne communication entre les périphériques, le masque de sous-réseau configuré sur le PC exécutant l'application TwidoSoft et celui de l'automate Twido doivent correspondre.</p> <p><b>Remarque :</b> Utilisez le masque de sous-réseau par défaut, sauf si votre automate Twido a des exigences particulières en matière de sous-réseau.</p>
Passerelle	<p>Saisissez dans ce champ l'adresse IP de la passerelle. Sur le réseau LAN, la passerelle doit se trouver sur le même segment que l'automate Twido. En règle générale, cette information vous est fournie par votre administrateur réseau.</p> <p>Notez que l'application ne fournit aucune valeur par défaut ; vous devez renseigner ce champ avec une adresse de passerelle valide.</p> <p><b>Remarque :</b> Si aucune passerelle n'est reliée au réseau, saisissez simplement l'adresse IP de votre automate Twido dans le champ Passerelle.</p>



## Onglet IP repérée

### Présentation

Ce sous-chapitre décrit la configuration de l'onglet IP repérée de la boîte de dialogue Configuration Ethernet.

**Note :**

- Cet onglet peut être configuré uniquement lorsque le programme d'application TwidoSoft est en mode local.
- Vous ne pouvez utiliser l'adresse IP repérée que si vous avez configuré manuellement l'adresse IP de l'automate Twido dans l'onglet Configurer adresse IP. L'adresse IP repérée ne fonctionne pas avec l'adresse IP par défaut.

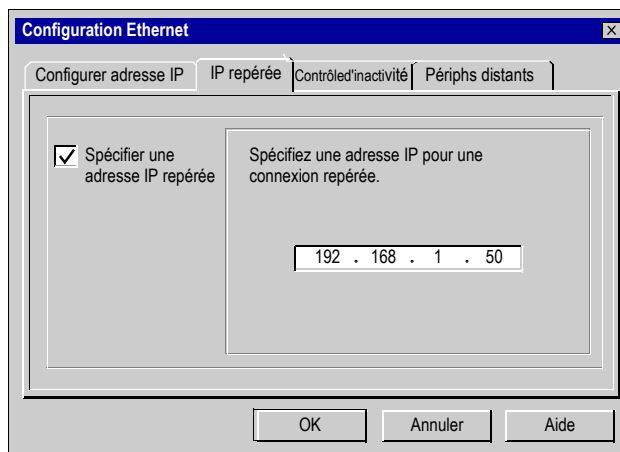
### Définition de la fonction d'adresse IP repérée

Cette fonction permet de réserver une des quatre voies de connexion TCP Ethernet prises en charge par l'automate Twido à un hôte client particulier appelé adresse IP repérée.

Cette adresse garantit qu'une voie TCP est réservée et toujours disponible pour une communication avec le périphérique distant spécifié, même si la durée d'inactivité est désactivée (définie sur "0").

### Onglet IP repérée

L'illustration suivante présente une capture de l'onglet IP repérée, contenant un exemple d'adresse IP repérée saisie par l'utilisateur :



**Configuration de  
l'onglet IP  
repérée**

Pour configurer l'onglet IP repérée, procédez comme suit :

Etape	Action
1	Cochez la case <b>Spécifier une adresse IP repérée</b> pour activer la fonction correspondante. Par défaut, l'adresse IP repérée est désactivée. <b>Résultat</b> : La zone d'adresse IP devient active dans la partie droite du cadre (cf. illustration précédente).
2	Saisissez l'adresse IP de l'hôte client dont vous voulez repérer l'IP dans la zone prévue à cet effet. <b>Remarque</b> : Par défaut, ce champ n'est pas renseigné. Vous devez définir l'adresse IP du périphérique repéré ou décocher la case Spécifier une adresse IP repérée pour désactiver cette fonction.

---

## Onglet Contrôle d'inactivité

### Présentation

Ce sous-chapitre décrit la configuration de l'onglet Contrôle d'inactivité de la boîte de dialogue Configuration Ethernet.

**Note :** Le contrôle d'inactivité de l'automate Twido peut être configuré uniquement lorsque le programme d'application TwidoSoft est en mode local.

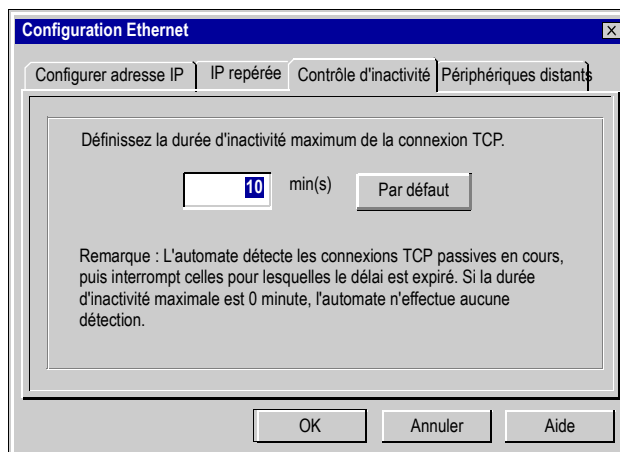
### Définition du contrôle d'inactivité

Le contrôle d'inactivité applique un délai d'inactivité à toutes les connexions TCP Ethernet courantes de l'automate Twido. Le délai d'inactivité correspond au temps pendant lequel une des quatre voies de connexion TCP Ethernet peut rester inactive avant que la connexion hôte client distante à cette voie ne soit interrompue.

**Remarque :** Le temporisateur d'inactivité est réinitialisé lorsqu'un trafic de données est détecté sur la voie de connexion surveillée.

### Onglet Contrôle d'inactivité

L'illustration suivante représente l'onglet Contrôle d'inactivité, avec la valeur par défaut de 10 min du temporisateur d'inactivité :



### Configuration de l'onglet Contrôle d'inactivité

Pour configurer le temporisateur d'inactivité, saisissez directement la durée en minutes dans la zone de texte **min(s)** comme indiqué sur la figure ci-dessus.

**Note :**

1. La durée par défaut est égale à 10 minutes. Après avoir saisi la valeur, si vous souhaitez **réinitialiser** la durée sur 10 minutes, cliquez sur le bouton **Par défaut**.
2. Pour **désactiver** la fonction Contrôle d'inactivité, définissez la durée sur **0**. L'automate Twido n'effectue plus de contrôle d'inactivité. Par conséquent, les connexions TCP sont conservées indéfiniment.
3. La durée d'inactivité maximum autorisée est égale à 255 minutes.

## Onglet Périphériques distants

### Présentation

Ce sous-chapitre décrit la procédure de configuration de l'onglet Périphériques distants de la boîte de dialogue Configuration Ethernet lorsque vous souhaitez utiliser l'instruction EXCH3 afin que l'automate Twido fonctionne en tant que client Modbus TCP/IP.

**Note :** L'onglet Périphériques distants de l'automate Twido peut être configuré uniquement lorsque le programme d'application TwidoSoft est en mode local.

### Informations préalables

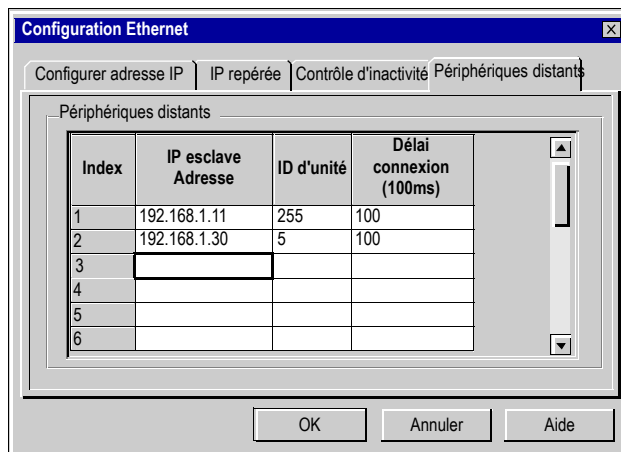
Il n'est pas nécessaire de configurer les périphériques distants sur les automates autres que celui avec lequel vous souhaitez utiliser l'instruction (EXCH3) du client Modbus TCP/IP (maître Modbus hérité).

### Tableau des périphériques distants

Le tableau des périphériques distants contient des informations relatives aux automates distants (fonctionnant en tant que serveurs Modbus TCP/IP) sur un réseau Ethernet qui peut être interrogé par le client Modbus TCP/IP via l'instruction EXCH3. Vous devez donc configurer le tableau des périphériques distants de façon que l'automate client Modbus TCP/IP puisse interroger les automates serveur Modbus TCP/IP sur le réseau.

### Onglet Périphériques distants

L'illustration suivante représente l'onglet Périphériques distants configuré de l'automate Twido fonctionnant en tant que client Modbus TCP/IP.



**Configuration de l'onglet Périphériques distants**

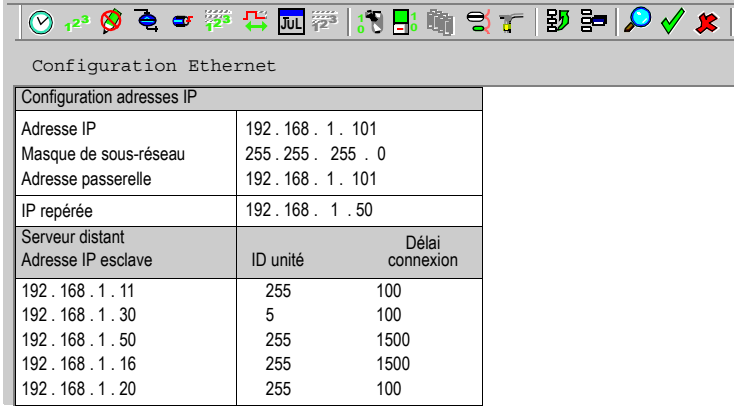
Les informations suivantes expliquent comment configurer les différents champs de l'onglet Périphériques distants :

Champ	Configuration
Index	<p>Ce champ en lecture seule contient l'index du protocole d'application Modbus (MBAP - Modbus Application Protocol) associé à l'adresse IP du réseau Ethernet du périphérique distant (serveur Modbus TCP/IP spécifié dans le champ Adresse IP esclave). L'instruction EXCH3 appelle l'index MBAP comme l'un des arguments de la fonction afin d'identifier quel automate distant spécifié dans le tableau est interrogé par le client Modbus TCP/IP.</p> <p><b>Remarque :</b> Vous pouvez spécifier jusqu'à 16 périphériques distants différents qui sont indexés de 1 à 16 dans ce tableau.</p>
Adresse IP esclave	<p>Saisissez l'adresse IP du périphérique distant (serveur Modbus TCP/IP) dans ce champ.</p> <p><b>Remarque :</b> Vous devez configurer les adresses IP esclaves consécutivement, dans l'ordre croissant et en commençant par l'index 1. Par exemple, vous ne pouvez pas configurer l'IP esclave de l'index 3 après celui de l'index 1. Vous devez auparavant configurer l'index 2.</p>
ID unité	<p>Saisissez l'ID de l'unité Modbus (ou adresse de protocole) dans ce champ. La plage d'ID d'unité doit être comprise entre 0 et 255. Le paramètre par défaut est 255.</p> <p>Un ID d'unité (différent de 255) permet la communication avec un périphérique distant via un pont ou une passerelle Modbus. Si le périphérique cible est un autre automate Twido ou un périphérique Modbus hérité installé sur un autre bus (adresse de liaison série via une passerelle), vous pouvez alors définir l'ID d'unité de ce périphérique distant en conséquence.</p> <p>Dans ce champ, vous devez définir l'adresse IP esclave en tant qu'adresse IP de la passerelle ou du pont et définir l'ID d'unité en tant qu'adresse de liaison série Modbus de votre périphérique cible.</p>
Délai connexion (100 ms)	<p>Spécifiez la durée (par 100 ms) pendant laquelle l'automate Twido tente d'établir une connexion TCP avec le périphérique distant. Si la connexion n'est toujours pas établie après l'expiration du délai, l'automate Twido ne tente plus de se connecter jusqu'à la prochaine requête de connexion via une instruction EXCH3.</p> <p>La plage valide de délai est comprise entre 0 et 65 535 (soit entre 0 et 6 553,5 s) Le paramètre par défaut est 100.</p>

## Affichage de la configuration Ethernet

**Présentation** Utilisez l'**éditeur de configuration** de TwidoSoft pour afficher la configuration Ethernet actuelle de l'automate Twido.

**Affichage de la configuration Ethernet** Pour afficher les paramètres de configuration Ethernet actuels à l'aide de l'éditeur de configuration, procédez comme suit :

Etape	Action
1	Sélectionnez <b>Programme &gt; Editeur de configuration</b> dans la barre de menus TwidoSoft.
2	Cliquez sur le raccourci <b>ETH</b> dans la barre des tâches de l'éditeur de configuration ou cliquez deux fois sur le raccourci <b>Port Ethernet</b> du Navigateur application.
3	Les paramètres de configuration TCP/IP Ethernet apparaissent dans un tableau de la manière suivante : 

Etape	Action
4	<p>A ce stade, si vous venez d'apporter des modifications aux paramètres de configuration TCP/IP Ethernet de votre automate Twido, vous pouvez soit les valider, soit les ignorer et restaurer la configuration précédente en procédant comme suit :</p> <ul style="list-style-type: none"><li>● Sélectionnez <b>Outils &gt; Accepter les modifications</b> dans la barre de menus TwidoSoft pour conserver les modifications apportées à la configuration TCP/IP Ethernet.</li><li>● Sélectionnez <b>Outils &gt; Annuler les modifications</b> pour ignorer les modifications et restaurer les paramètres de la configuration TCP/IP Ethernet précédente.</li><li>● Sélectionnez <b>Outils &gt; Editer...</b> pour revenir à la boîte de dialogue Configuration Ethernet et modifier les paramètres de configuration TCP/IP.</li><li>● Sélectionnez <b>Outils &gt; Mettre à jour le programme automate</b> pour charger la totalité du fichier de configuration automate dans l'automate Twido.</li></ul>

---



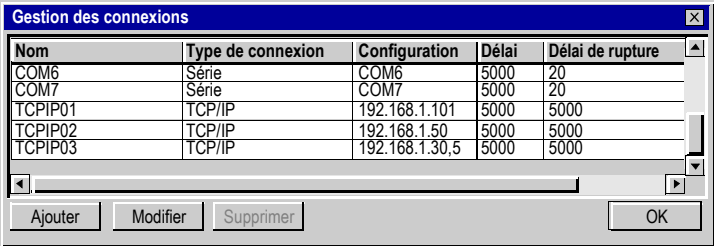
## Gestion des connexions Ethernet

### Présentation

Ce sous-chapitre explique comment configurer/ajouter/supprimer/sélectionner une connexion TCP/IP Ethernet entre un PC et un automate.

### Configuration d'une nouvelle connexion TCP/IP

Suivez les instructions ci-dessous pour configurer une connexion TCP/IP Ethernet entre le PC exécutant l'application TwidoSoft et un automate TWDLCAE40DRF installé sur votre réseau.

Etape	Action
1	<p>Sélectionnez <b>Fichier &gt; Préférences &gt; Gestion des connexions</b> dans la barre de menus TwidoSoft pour afficher la boîte de dialogue Gestion des connexions illustrée ci-dessous :</p> 
2	<p>Cliquez sur le bouton <b>Ajouter</b> dans la boîte de dialogue Gestion des connexions.</p> <p><b>Résultat :</b> Une ligne de connexion supplémentaire est ajoutée. Elle comprend les paramètres de connexion par défaut conseillés. Vous devez modifier ces paramètres.</p> <p><b>Remarque :</b> Pour définir une nouvelle valeur dans un champ, vous avez deux possibilités :</p> <ul style="list-style-type: none"><li>● Cliquez sur le champ voulu, puis sur le bouton <b>Modifier</b>.</li><li>● Cliquez deux fois dans le champ voulu.</li></ul>
3	<p>Dans le champ <b>Nom</b>, saisissez un nom descriptif pour la nouvelle connexion. Un nom valide contient au maximum 32 caractères alphanumériques.</p>
4	<p>Cliquez dans le champ <b>Type de connexion</b> pour dérouler la liste et sélectionnez <b>TCP/IP</b> puisque vous configurez une nouvelle connexion Ethernet entre un PC et un automate Twido prenant en charge Ethernet.</p>

Etape	Action
5	<p>Dans le champ <b>Configuration</b>, saisissez une adresse IP et un ID d'unité valides (le cas échéant) qui correspondent aux informations IP de l'automate Twido TWDLCAE40DRF auquel vous souhaitez vous connecter. L'adresse IP et l'ID d'unité doivent être séparés par une virgule.</p> <p><b>Adresse IP</b> : Selon la solution de configuration de l'automate Twido choisie, saisissez soit l'adresse IP par défaut, soit l'adresse IP statique spécifique à l'utilisateur affectée à l'automate.</p> <p><b>ID d'unité</b> : Saisissez un entier compris entre 0 et 255.</p> <ul style="list-style-type: none"> <li>● Si l'automate Twido cible est situé après une passerelle ou un pont sur une liaison série Modbus, l'ID d'unité correspond au repère série du périphérique.</li> <li>● Si l'automate Twido cible est situé sur la même couche réseau Ethernet que votre PC, vous n'êtes pas obligé de renseigner ce champ. L'ID d'unité par défaut (255) est affecté automatiquement.</li> </ul>
6	<p>Dans le champ <b>Délai</b>, saisissez une valeur de délai en millisecondes (ms) pour établir une connexion avec l'automate Twido. Après expiration du délai, si le PC n'a pas réussi à se connecter à l'automate, l'application TwidoSoft ne tente plus d'établir la connexion. Pour reprendre les tentatives de connexion, sélectionnez <b>Automate &gt; Sélectionner une connexion</b> dans la barre de menus TwidoSoft.</p> <p><b>Remarque</b> :La valeur maximum du délai est 65 535 ms (65,5 s).</p>
7	<p>Le <b>Délai de rupture</b> est la durée maximum autorisée entre la requête Modbus TCP/IP et la réception de la trame de la réponse. Si le délai avant rupture est dépassé sans réception de la trame de réponse requise, l'application TwidoSoft interrompt la connexion entre le PC et l'automate.</p> <p>Remarque : La valeur maximum du délai est 65 535 ms (65,5 s). La valeur par défaut est 5 000 ms. La valeur 0 n'est pas une entrée valide, vous devez renseigner ce champ avec une valeur différente de zéro.</p> <p><b>Remarque</b> :La valeur maximum du délai est 65 535 ms (65,5 s).</p>
8	<p>Cliquez sur <b>OK</b> pour enregistrer les nouveaux paramètres de connexion et fermer la boîte de dialogue Gestion des connexions.</p> <p><b>Résultat</b> : Les noms de toutes les nouvelles connexions sont ajoutés à la liste déroulante des connexions dans la boîte de dialogue <b>Fichier &gt; Préférences</b> et dans <b>Automate &gt; Sélectionner une connexion</b>.</p>

**Modification et suppression d'une connexion TCP/IP**

Pour supprimer ou modifier les paramètres des connexions TCP/IP Ethernet existantes, procédez comme suit :

- Pour supprimer une connexion de la boîte de dialogue Gestion Ethernet, cliquez sur le nom de la connexion appropriée, puis cliquez sur le bouton **Supprimer**. Une fois la connexion supprimée, tous les paramètres correspondants sont définitivement perdus.
  - Pour modifier les paramètres d'une connexion existantes, sélectionnez le champ approprié, puis cliquez sur le bouton **Modifier**. Vous pouvez ensuite saisir la nouvelle valeur dans le champ sélectionné.
-

## Voyants Ethernet

### Présentation

Deux voyants de communication Ethernet se trouvent sur le panneau d'affichage des voyants situé sur la face avant de l'automate TWDLCAE40DRF. Ils sont également représentés dans l'application TwidoSoft sous **Automate > Vérifier l'automate**. Ils se nomment de la façon suivante :

- LAN ACT
- LAN ST

Les voyants Ethernet permettent de surveiller de manière continue l'état et le diagnostic des connexions du port Ethernet.

### Etat des voyants

Le tableau suivant présente l'état des voyants Ethernet **LAN ACT** et **LAN ST**.

Voyant	Etat	Couleur	Description
LAN ACT	Eteint	-	Aucun signal Ethernet sur le port RJ-45.
	Allumé en continu	Vert	Signal de battement de liaison 10BASE-TX indiquant une connexion de 10 Mbit/s.
	Clignotant		Paquets de données envoyés ou reçus via une connexion 10BASE-TX.
	Allumé en continu	Orange	Signal de battement de liaison 100BASE-TX indiquant une connexion de 100 Mbit/s.
	Clignotant		Paquets de données envoyés ou reçus via une connexion 100BASE-TX.

Voyant	Etat	Couleur	Description
<b>LAN ST</b>	Allumé en continu	Vert	Base automate sous tension. Le port Ethernet est prêt à communiquer sur le réseau.
	2 clignotements		Initialisation d'Ethernet lors de la mise sous tension.
	2 clignotements, puis déconnexion		Aucune adresse MAC valide.
	3 clignotements, puis déconnexion		Il existe trois causes possibles : <ul style="list-style-type: none"> <li>● Aucun battement de liaison détecté.</li> <li>● Le câble réseau Ethernet n'est pas branché correctement ou est défectueux.</li> <li>● Le périphérique réseau (concentrateur/commutateur) est défectueux ou n'est pas correctement configuré.</li> </ul>
	4 clignotements, puis déconnexion		Adresse IP double détectée sur le réseau. (Pour y remédier, essayez d'affecter une nouvelle adresse IP à l'automate Twido.)
	6 clignotements, puis déconnexion		Utilisation d'une adresse IP par défaut valide et convertie ; mode FDR sûr.
	9 clignotements, puis déconnexion		Panne matérielle d'Ethernet.

## Messagerie Modbus TCP

---

### Présentation

Vous pouvez utiliser la messagerie Modbus TCP pour permettre au client Modbus TCP (automate maître) d'envoyer des messages Ethernet vers le serveur Modbus TCP (automate esclave) et d'en recevoir. Modbus TCP étant un protocole de communication poste à poste, un automate Twido prenant en charge Ethernet peut être client ou serveur selon qu'il envoie des requêtes ou qu'il y répond.

### Echange de messages sur le réseau Ethernet

L'échange de messages sur Ethernet est géré par l'instruction EXCH3 et le bloc fonction %MSG3 : le routage vers un hôte Ethernet ou via une passerelle est également pris en charge par EXCH3.

- **Instruction EXCH3** : pour émettre/recevoir des messages.
- **Bloc fonction %MSG3** : pour contrôler les échanges de messages.

### Instruction EXCH3

L'instruction EXCH3 permet à l'automate Twido d'envoyer et/ou recevoir des informations vers/depuis des nœuds du réseau Ethernet. L'utilisateur définit une table de mots (%MWi:L) contenant des informations de contrôle, ainsi que les données à envoyer et/ou recevoir (jusqu'à 128 octets en émission et/ou réception). Le format des tables de mots fait l'objet d'une description détaillée dans la section suivante.

Un échange de messages s'effectue à l'aide de l'instruction EXCH3 :

Syntaxe : [EXCH3 %MWi:L]

où : L = nombre de mots dans les tables de mots de commande,  
d'émission et de réception

L'automate Twido doit terminer l'échange de la première instruction EXCH3 avant de pouvoir en lancer un second. Le bloc fonction %MSG3 doit être utilisé lors de l'envoi de plusieurs messages.

Le traitement de l'instruction EXCH3 en langage liste d'instructions se produit immédiatement, toutes les émissions étant démarrées sous contrôle d'interruptions (la réception des données est également sous contrôle d'interruptions). Ce traitement est considéré comme un traitement en arrière-plan.

**Note** : L'instruction EXCH3 s'utilise de la même manière que l'instruction EXCHx (où x = 1 ou 2) utilisée avec un Modbus hérité. Les syntaxes de ces instructions sont également identiques. Cependant, il existe une différence essentielle concernant les informations contenues dans l'octet 1 des tables d'émission et de réception. Alors que l'octet 1 du Modbus hérité transporte l'adresse de liaison série de l'automate esclave, l'octet 1 du Modbus TCP transporte le numéro d'**index** de l'automate client Modbus TCP. Le numéro d'index est défini et stocké dans le tableau des périphériques distants de la boîte de dialogue Configuration Ethernet de TwidoSoft (pour plus d'informations, voir *Onglet Périphériques distants*, p. 181).

**Table de mots  
EXCH3**

La taille maximale des trames émises et/ou reçues est de 128 octets (notez que cette limite s'applique uniquement au client Modbus TCP, car le serveur Modbus TCP prend en charge la longueur de PDU Modbus standard de 256 octets). En outre, la table de mots associée à l'instruction EXCH3 se compose des tables de contrôle, d'émission et de réception, comme décrit ci-dessous :

	Octet de poids fort	Octet de poids faible
Table de contrôle	Commande	Longueur (Emission/Réception)
	Décalage réception	Décalage émission
Table d'émission	Octet 1 émis ( <b>Index</b> comme indiqué dans le tableau des périphériques distants de la boîte de dialogue Configuration Ethernet de TwidoSoft)	Octet 2 émis comme Modbus série
	...	...
	...	Octet n émis
	Octet n+1 émis	
Table de réception	Octet 1 reçu ( <b>Index</b> comme indiqué dans le tableau des périphériques distants de la boîte de dialogue Configuration Ethernet de TwidoSoft)	Octet 2 reçu comme Modbus série
	...	...
	...	Octet p reçu
	Octet p+1 reçu	

**Bloc fonction  
%MSG3**

La fonction %MSG3 s'utilise de la même manière que la fonction %MSGx utilisée avec le Modbus hérité. Elle permet de gérer les échanges de données de la manière suivante :

- Vérification des erreurs de communication
- Coordination des messages multiples
- Emission de messages prioritaires

Le bloc fonction %MSGx dispose d'une entrée et de deux sorties associées :

Entrée/Sortie	Définition	Description
R	Entrée RAZ	Mise à 1 : réinitialise la communication ou le bloc (%MSGx.E = 0 et %MSGx.D = 1)
%MSGx.D	Communication terminée	0 : requête en cours 1 : communication terminée en cas de fin d'émission, de réception du caractère de fin, d'erreur ou de réinitialisation du bloc
%MSGx.E	Erreur	0 : longueur du message et liaison corrects 1 : en cas de mauvaise commande, de table configurée de manière incorrecte, de mauvais caractère reçu (vitesse, parité, etc.) ou de saturation de la table de réception



## Code d'erreur EXCH3

Lorsqu'une erreur survient avec l'instruction EXCH3 :

- les bits **%MSG3.D** et **%MSG3.E** sont mis à **1**, et
- le **code d'erreur** de la communication Ethernet est enregistré dans le mot système **%SW65**.

Le tableau suivant présente le code d'erreur EXCH3 :

<b>Code d'erreur EXCH3 (enregistré dans le mot système %SW65)</b>
<p><b>Codes d'erreur standard communs à toutes les instructions EXCHx (x = 1, 2, 3) :</b></p> <ul style="list-style-type: none"> <li>0 - opération réussie</li> <li>1 - nombre d'octets à émettre trop important (&gt; 128)</li> <li>2 - table d'émission trop petite</li> <li>3 - table de mots trop petite</li> <li>4 - débordement de la table de réception</li> <li>5 - délai écoulé (Remarque : le code d'erreur 5 est invalidé par l'instruction EXCH3 et remplacé par les codes d'erreur 109 et 122 spécifiques à Ethernet qui sont décrits ci-dessous.)</li> <li>6 - émission</li> <li>7 - mauvaise commande dans la table</li> <li>8 - port sélectionné non configuré/disponible</li> <li>9 - erreur de réception</li> <li>10 - impossible d'utiliser %KW en cas de réception</li> <li>11 - décalage d'émission plus important que la table d'émission</li> <li>12 - décalage de réception plus important que la table de réception</li> <li>13 - interruption du traitement EXCH par l'automate</li> </ul>
<p><b>Codes d'erreur dédiés à la réponse Modbus :</b></p> <ul style="list-style-type: none"> <li>81 - l'automate esclave (serveur) renvoie la réponse FONCTION INCORRECTE</li> <li>82 - l'automate esclave (serveur) renvoie la réponse ADRESSE DE DONNEES INCORRECTE</li> <li>83 - l'automate esclave (serveur) renvoie la réponse VALEUR DE DONNEES INCORRECTE</li> <li>84 - l'automate esclave (serveur) renvoie la réponse ERREUR PERIPHERIQUE ESCLAVE</li> <li>85 - l'automate esclave (serveur) renvoie la réponse CONFIRMATION</li> <li>86 - l'automate esclave (serveur) renvoie la réponse PERIPHERIQUE ESCLAVE OCCUPE</li> <li>87 - l'automate esclave (serveur) renvoie la réponse CONFIRMATION NEGATIVE</li> <li>88 - l'automate esclave (serveur) renvoie la réponse ERREUR DE PARITE MEMOIRE</li> </ul>

<b>Code d'erreur EXCH3 (enregistré dans le mot système %SW65)</b>
---

**Codes d'erreur Ethernet pour EXCH3 :**

- 101 - aucune adresse IP de ce type
- 102 - la connexion TCP est interrompue
- 103 - aucun socket disponible (toutes les voies de connexion sont occupées)
- 104 - le réseau ne fonctionne pas
- 105 - le réseau est inaccessible
- 106 - le réseau a interrompu la connexion lors de la réinitialisation
- 107 - la connexion a été abandonnée par le poste
- 108 - la connexion a été réinitialisée par le poste
- 109 - délai écoulé pour la connexion
- 110 - rejet de la tentative de connexion
- 111 - l'hôte ne fonctionne pas
- 120 - index inconnu (le périphérique distant n'est pas indexé dans le tableau de configuration)
- 121 - erreur fatale (MAC, puce, adresse IP double)
- 122 - délai de réception écoulé après l'envoi des données
- 123 - initialisation d'Ethernet en cours

---

# Fonctions analogiques intégrées

# 7

---

## Présentation

### Objet de ce chapitre

Cette rubrique décrit la gestion de la voie analogique et des potentiomètres analogiques intégrés.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Point de réglage analogique	196
Voie analogique	198

## Point de réglage analogique

---

### Introduction

Les automates Twido possèdent :

- un point de réglage analogique sur les automates TWDLC•A10DRF et TWDLC•A16DRF et sur tous les automates modulaires (TWDLMDA20DTK, TWDLMDA20DUK, TWDLMDA20DRT, TWDLMDA40DTK et TWDLMDA40DUK),
- deux points de réglage sur les automates TWDLC•A42DRF et TWDLCA•40DRF.

### Programmation

Les valeurs numériques, allant de 0 à 1 023 pour le point de réglage analogique 1 et de 0 à 511 pour le point de réglage analogique 2, correspondant aux valeurs analogiques données par ces points de réglage analogiques sont contenues dans les deux mots d'entrée suivants :

- %IW0.0.0 pour le point de réglage analogique 1 (situé à gauche)
- %IW0.0.1 pour le point de réglage analogique 2 (situé à droite)

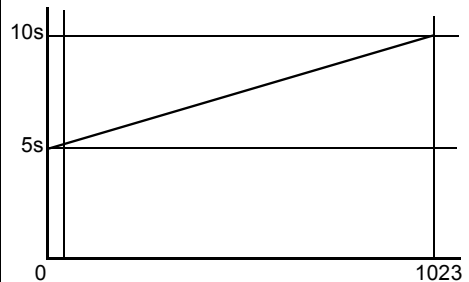
Ces mots peuvent être utilisés dans les opérations arithmétiques et pour n'importe quel type de réglage (présélection d'une temporisation ou d'un compteur, ajustement de la fréquence du générateur d'impulsions ou de la durée de préchauffage d'une machine, etc.).

---

**Exemple**

Utilisation du point de réglage analogique 1 pour modifier la durée de temporisation de 5 à 10 secondes :

Ce réglage utilise la quasi-totalité de la plage du point de réglage analogique 1 (0 à 1 023).

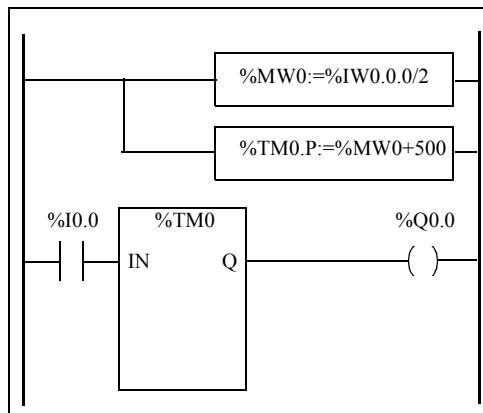


Les paramètres suivants sont sélectionnés au moment de la configuration du bloc de temporisation %TM0 :

- Type TON
- Base temps : 10 ms

La valeur de présélection de la durée de temporisation est calculée à partir de la valeur du point de réglage analogique, à l'aide de l'équation suivante :  $\%TM0.P := (\%IW0.0/2)+500$ .

Code pour l'exemple précédent :



```
LD      1
[%MW0:=%IW0.0/2]
[%TM0.P:=%MW0+500]
BLK    %TM0
LD      %I0.0
IN
OUT_BLK
LD      Q
ST      %Q0.0
END_BLK
.....
```

Voie analogique

Introduction

Tous les automates modulaires (TWDLMDA20DTK, TWDLMDA20DUK, TWDLMDA20DRT, TWDLMDA40DTK et TWDLMDA40DUK) possèdent une voie analogique. La tension en entrée est comprise entre 0 et 10 V et entre 0 et 511 pour le signal numérisé. La voie analogique utilise un schéma de calcul de moyennes simple qui s'applique sur huit échantillons.

Principe

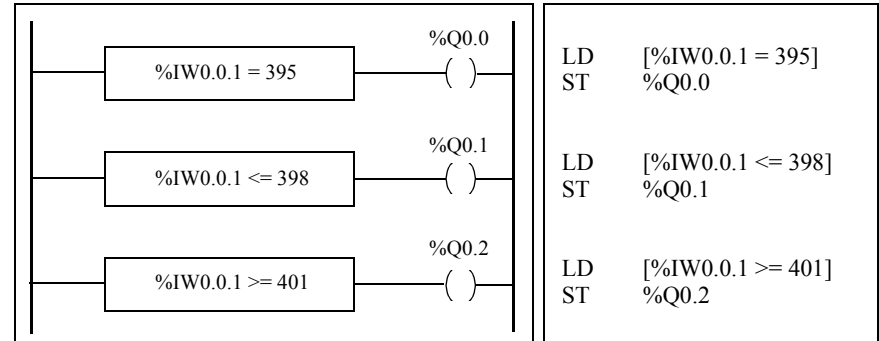
Un convertisseur de données analogiques en données numériques échantillonne une tension comprise entre 0 et 10 V en une valeur numérique comprise entre 0 et 511. Cette valeur est stockée dans le mot système %IW0.0.1. La valeur est linéaire sur l'intégralité de la plage, et chaque incrément est de 20 mV (10 V/512). Dès que le système détecte la valeur 511, la voie est considérée comme saturée.

Exemple de programmation

**Régulation de la température d'un four :** La température de cuisson est réglée sur 350°C. Une variation de +/- 2,5°C engendre une disjonction des sorties %Q0.0 et %Q0.2. La quasi-totalité de la plage de paramètres possibles de la voie analogique (de 0 à 511) est utilisée dans cet exemple. Les paramètres analogiques des différentes températures sont les suivants :

Température (°C)	Tension	Mot système %IW0.0.1
0	0	0
347,5	7,72	395
350	7,77	398
352,5	7,83	401
450	10	511

Code pour l'exemple précédent :



---

# Gestion des modules analogiques

## 8

---

### Présentation

#### Objet de ce chapitre

Ce chapitre offre une présentation des procédures de gestion des modules analogiques des automates Twido.

#### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation des modules analogiques	200
Adressage d'entrées et de sorties analogiques	201
Configuration d'entrées et de sorties analogiques	203
Informations sur l'état du module analogique	205
Exemples d'utilisation de modules analogiques	206

---

## Présentation des modules analogiques

### Introduction


Outre le potentiomètre analogique 10 bits et la voie analogique 9 bits, l'ensemble des automates Twido prenant en charge l'expansion d'E/S sont également capables de communiquer avec des modules d'E/S analogiques.

Ces modules analogiques sont les suivants :

Nom	Voies	Plage du signal	Codage
TWDAMI2HT	2 entrée	0 - 10 V ou 4 - 20 mA	12 bits
TWDAM01HT	1 sortie	0 - 10 V ou 4 - 20 mA	12 bits
TWDAMM3HT	2 entrée, 1 sortie	0 - 10 V ou 4 - 20 mA	12 bits
TWDALM3LT	2 entrée, 1 sortie	0 - 10 V, Entrées Th ou PT100, Sorties de 4 à 20 mA	12 bits

### Fonctionnement des modules analogiques

Les mots en entrée et en sortie (%IW et %QW) sont utilisés pour échanger des données entre l'application utilisateur et les voies analogiques. La mise à jour de ces mots est effectuée de manière synchronisée avec la scrutation de l'automate en mode RUN.

	<b>ATTENTION</b>
	<b>Mise en route non désirée d'équipements</b>
	Lorsque l'automate est en position STOP, la sortie analogique se trouve en position de repli. Dans le cas d'une sortie numérique, la position de repli est zéro.  <b>Le non-respect de ces précautions peut entraîner des lésions corporelles ou des dommages matériels.</b>



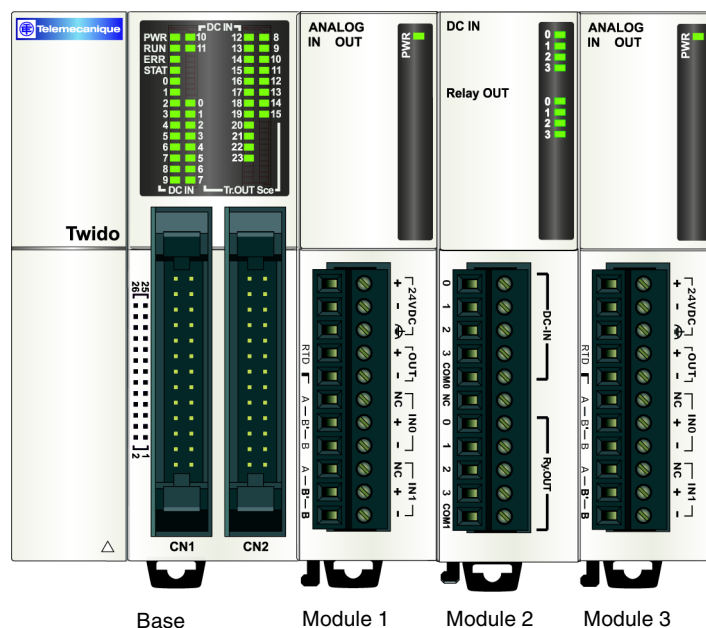
## Adressage d'entrées et de sorties analogiques

### Introduction

Des adresses sont affectées aux voies analogiques en fonction de leur emplacement sur le bus d'expansion.

### Exemple d'adressage d'E/S analogique

Dans cet exemple, un module TWDLMDA40DUK possède un potentiomètre de réglage analogique 10 bits intégré, ainsi qu'une voie analogique 9 bits intégrée. Sur le bus d'expansion, sont configurés : un module analogique TWDAMM3HT, un module de relais numérique d'E/S TWDDMM8DRT, ainsi qu'un second module analogique TWDAMM3HT.



Le tableau suivant présente une description détaillée de l'adressage de chaque sortie.

Description	Base	Module 1	Module 2	Module 3
Potentiomètre analog. 1	%IW0.0.0			
Voie analogique intégrée	%IW0.0.1			
Voie 1 d'entrée analogique		%IW0.1.0		%IW0.3.0
Voie 2 d'entrée analogique		%IW0.1.1		%IW0.3.1
Voie 1 de sortie analogique		%QW0.1.0		%QW0.3.0
Voies d'entrée TOR			%I0.2.0 - %I0.2.3	
Voies de sortie TOR			%Q0.2.0 - %Q0.2.3	

---

## Configuration d'entrées et de sorties analogiques

### Introduction

Cette section présente des informations sur la configuration des entrées et des sorties du module analogique.

### Configuration d'E/S analogique

La boîte de dialogue Configurer un module permet de gérer les paramètres des modules analogiques.

**Note :** Vous pouvez modifier ces paramètres seulement en mode local, lorsque vous n'êtes pas connecté à un automate.


Des adresses sont affectées aux voies analogiques en fonction de leur emplacement sur le bus d'expansion. Afin de vous aider durant la phase de programmation et de faciliter l'utilisation des données dans votre application utilisateur vous pouvez également utiliser des symboles prédéfinis.

Vous pouvez configurer la voie de sortie unique de TWDAM01HT, TWDAMM3HT TWDALM3LT comme suit :

- Non utilisé
- 0 – 10 V
- 4 – 20 mA

Vous pouvez configurer les deux voies d'entrée de TWDAMI2HT et TWDAMM3HT comme suit :

- Non utilisé
- 0 – 10 V
- 4 – 20 mA

	<b>ATTENTION</b>
	<p><b>Risque de détérioration du matériel</b></p> <p>Si vous raccordez une entrée tension sur un module qui a été configuré à l'aide de TwidoSoft en courant, vous risquez d'endommager le module analogique de façon irréversible. Assurez-vous que le raccordement est conforme à la configuration TwidoSoft.</p> <p><b>Le non-respect de ces précautions peut entraîner des lésions corporelles ou des dommages matériels.</b></p>

Les deux voies d'entrée de TWDALM3LT peuvent être configurées comme suit :

- Non utilisé
- Thermocouple K
- Thermocouple J
- Thermocouple T
- PT 100

Lorsqu'une voie est configurée, vous pouvez lui affecter des unités et définir la plage des entrées en fonction du tableau suivant.

Plage	Unités	Description
Normale	Aucune	Plage fixe allant de 0 à 4095 (valeurs minimale et maximale).
Personnalisée	Aucune	Définie par l'utilisateur, mais comprise entre -32 768 et +32 767.
Celsius	0.1°C	Echelle thermométrique internationale. Uniquement disponible pour les voies d'entrée TWDALM3LT.
Fahrenheit	0.1°F	Echelle thermométrique dans laquelle le point d'ébullition de l'eau est fixé à 212°F (100°C) et le point de gel à 32°F (0°C). Uniquement disponible pour les voies d'entrée TWDALM3LT.

---

## Informations sur l'état du module analogique

**Tableau d'état** Le tableau suivant contient les informations nécessaires pour contrôler l'état des modules d'E/S analogique.

Mot système	Fonction	Description
%SW80	Etat de l'E/S de base	Bit [0] Voies en fonctionnement normal (pour toutes ses voies). Bit [1] Module en cours d'initialisation (ou initialisation des informations de toutes les voies). Bit [2] Défaut matériel (défaut d'alimentation externe, commun à toutes les voies) Bit [3] Défaut de configuration de l'automate Bit [4] Conversion de la voie d'entrée des données 0 en cours Bit [5] Conversion de la voie d'entrée des données 1 en cours Bit [6] Voie thermocouple d'entrée 0 non configurée Bit [7] Voie thermocouple d'entrée 1 non configurée Bit [8] Non utilisé Bit [9] Non utilisé Bit [10] Voie des données d'entrée analogique 0 au dessus de la plage Bit [11] Voie des données d'entrée analogique 1 au dessus de la plage Bit [12] Liaison incorrecte (voie des données d'entrée analogique 0 au-dessous de la plage courante, boucle de courant ouverte) Bit [13] Liaison incorrecte (voie des données d'entrée analogique 1 au-dessous de la plage courante, boucle de courant ouverte) Bit [14] Non utilisé Bit [15] Voie de sortie non disponible
%SW81	Module d'expansion d'E/S 1	Etat : Définitions identiques à %SW80
%SW82	Module d'expansion d'E/S 2	Etat : Définitions identiques à %SW80
%SW83	Module d'expansion d'E/S 3	Etat : Définitions identiques à %SW80
%SW84	Module d'expansion d'E/S 4	Etat : Définitions identiques à %SW80
%SW85	Module d'expansion d'E/S 5	Etat : Définitions identiques à %SW80
%SW86	Module d'expansion d'E/S 6	Etat : Définitions identiques à %SW80
%SW87	Module d'expansion d'E/S 7	Etat : Définitions identiques à %SW80

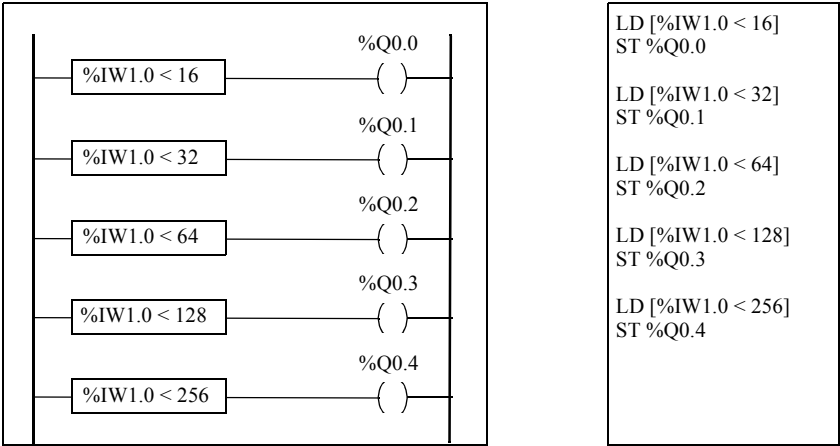
## Exemples d'utilisation de modules analogiques

### Introduction

Ce sous-chapitre présente un exemple d'utilisation des modules analogiques des automates Twido.

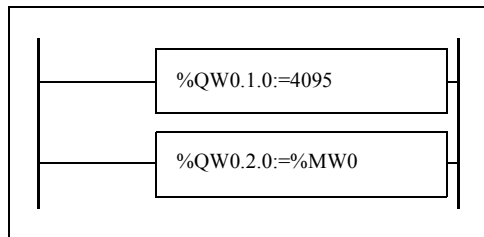
### Exemple : entrée analogique

Cet exemple compare le signal d'entrée analogique avec cinq valeurs de seuil distinctes. Une comparaison de l'entrée analogique est effectuée et un bit est réglé sur la base automate si le signal d'entrée est inférieur ou égal au seuil.



**Exemple : sortie analogique**

Dans le programme ci-dessous on utilise une carte analogique dans l'emplacement 1 et 2. La carte utilisée dans l'emplacement 1 a une sortie 10 volts avec la gamme "normal" :



```
LD 1
[%QW0.1.0:=4095
LD 1
[%QW0.2.0:=%MW0
```

- Exemple de valeurs de sorties pour %QW1.0=4095 (cas normal) :

Le tableau ci-dessous donne la valeur de la tension de sortie suivant la valeur maximale attribuée à %QW1.0 :

	valeur numérique	valeur analogique (volt)
Minimum	0	0
Maximum	4095	10
Valeur 1	100	0,244
Valeur 2	2460	6

- Exemple de valeurs de sorties pour pour une gamme personnalisée (minimum =0, maximum =1000) :

Le tableau ci-dessous donne la valeur de la tension de sortie suivant la valeur maximale attribuée à %QW1.0 :

	valeur numérique	valeur analogique (volt)
Minimum	0	0
Maximum	1000	10
Valeur 1	100	1
Valeur 2	600	6





---

# Mise en œuvre du bus AS-Interface V2

# 9

---

## Présentation

### Objet de ce chapitre

Ce chapitre fournit les informations sur la mise en œuvre logicielle du module maître AS-Interface TWDNOI10M3 et de ses esclaves.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Présentation du bus AS-Interface V2	210
Description fonctionnelle générale	211
Principes de mise en œuvre logicielle	214
Description de l'écran de configuration du bus AS-Interface	216
Configuration du bus AS-Interface	218
Description de l'écran de mise au point	224
Modification de l'adresse d'un esclave	227
Mise à jour de la configuration du bus AS-Interface en mode connecté	229
Adressage automatique d'un esclave AS-Interface V2	234
Comment insérer un équipement esclave dans une configuration AS-Interface V2 existante	235
Remplacement automatique d'un esclave AS-Interface V2 défectueux	236
Adressage des entrées/sorties associées aux équipements esclaves connectés sur bus AS-Interface V2	237
Programmation et diagnostic du bus AS-Interface V2	239
Mode de fonctionnement du module interface bus AS-Interface V2	244

## Présentation du bus AS-Interface V2

---

### Introduction

Le bus AS-Interface (Actuator Sensor-Interface) permet l'interconnexion, sur un câble unique, de capteurs/actionneurs au niveau le plus bas de l'automatisation. Ces capteurs/actionneurs seront définis dans la documentation comme **périphériques esclaves**.

La mise en œuvre de AS-Interface nécessite de définir le contexte physique de l'application dans laquelle il sera intégré (bus d'expansion, alimentation, processeur, modules, périphériques esclaves AS-Interface connectés sur le bus) puis d'en assurer sa mise en œuvre logicielle.

Cette mise en œuvre logicielle sera réalisée depuis les différents éditeurs de TwidoSoft :

- soit en mode local,
- soit en mode connecté.

### Bus AS-Interface V2

Le module maître AS-interface **TWDOI10M3** intègre les fonctionnalités suivantes :

- Profil M3 : ce profil offre toutes les fonctionnalités définies par la norme AS-Interface V2, mais ne prend pas en charge pas les profils analogiques S7-4.
- Une voie AS-Interface par module
- Repérage automatique de l'esclave avec le repère 0
- Gestion des profils et paramètres
- Protection contre l'inversion de polarité sur les entrées de bus

Le bus AS-Interface permet alors :

- jusqu'à 31 esclaves de type repérage standard et 62 de type repérage étendu,
- jusqu'à 248 entrées et 186 sorties,
- jusqu'à 7 esclaves analogiques (4 E/S max. par esclave),
- un temps de cycle de 10 ms maximum.

Deux modules maîtres AS-Interface maximum peuvent être connectés sur un automate modulaire Twido, un automate compact TWDLCA•A24DRF ou TWDLCA•40DRF.

---

## Description fonctionnelle générale

---

### Présentation générale

Pour la configuration AS-Interface, le logiciel TwidoSoft permet à l'utilisateur :

- de configurer le bus (déclaration des esclaves et attribution des adresses sur le bus) de façon manuelle,
- d'adapter la configuration par rapport à ce qui est présent sur le bus,
- de prendre en compte les paramètres des esclaves,
- de contrôler l'état du bus.

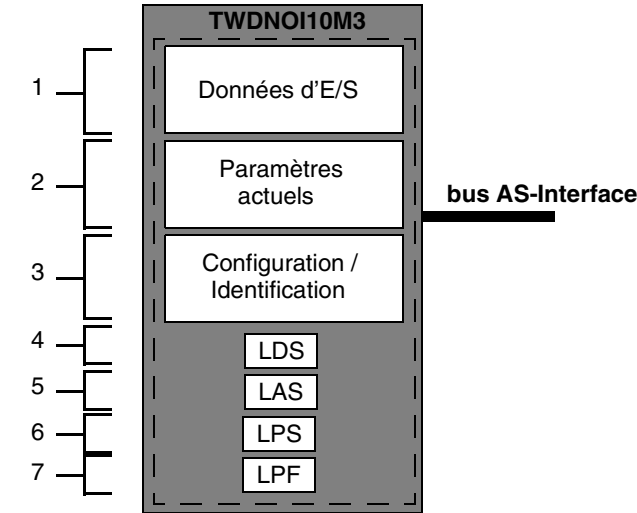
Pour cela toutes les informations en provenance ou à destination du maître AS-Interface sont stockées dans des objets (mots et bits) spécifiques.

---

Structure du maître AS-Interface

Le coupleur AS-Interface intègre des champs de données qui permettent de gérer des listes d'esclaves et les images des données d'entrées / sorties. Ces informations sont stockées en mémoire volatile.

La figure ci-dessous présente l'architecture du coupleur **TWDNOI10M3**.



Légende :

Repère	Élément	Description
1	Données d'E/S (IDI, ODI)	Images des 248 entrées et des 186 sorties du Bus AS-Interface V2.
2	Paramètres actuels (PI, PP)	Image des paramètres de tous les esclaves.
3	Configuration/ Identification (CDI, PCD)	Ce champ contient tous les codes E/S et les codes identification de tous les esclaves détectés.
4	LDS	Liste de tous les esclaves détectés sur le bus.
5	LAS	Liste des esclaves activés sur le bus.
6	LPS	Liste des esclaves prévus sur le bus et configurés par TwidoSoft.
7	LPF	Liste des esclaves ayant un défaut périphérique.

### Structure des équipements esclaves

Les esclaves en adressage standard disposent chacun de :

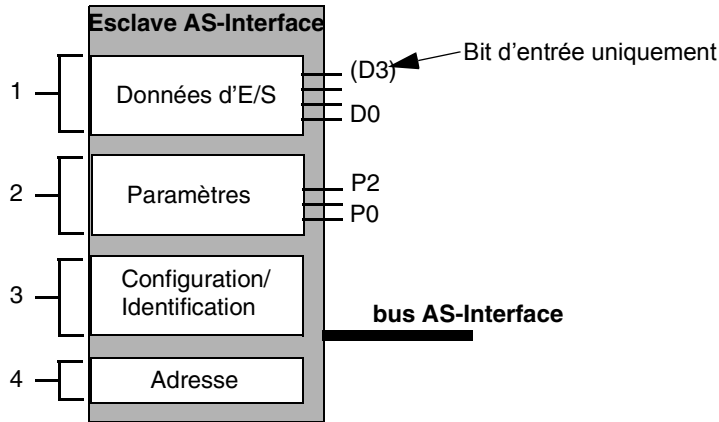
- 4 bits d'entrée/sortie,
- 4 bits de paramétrage.

Les esclaves en adressage étendu disposent chacun de :

- 4 bits d'entrée/sortie (dernier bit réservé à l'entrée uniquement),
- 3 bits de paramétrage.

Chaque esclave possède sa propre adresse, ainsi qu'un profil et sous-profil (définition de l'échange des variables).

La figure ci-dessous présente la structure d'un esclave en adressage étendu :



Légende :

Repère	Élément	Description
1	Données d'entrées/sorties	Les données d'entrées sont mémorisées par l'esclave et mises à la disposition du maître AS-Interface. Les données de sorties sont mises à jour par le coupleur maître.
2	Paramètres	Les paramètres permettent le pilotage et la commutation des modes de marche internes au capteur ou actionneur.
3	Configuration/ Identification	Ce champ contient : <ul style="list-style-type: none"> <li>• le code correspondant à la configuration des entrées/sorties (I/O),</li> <li>• le code d'identification de l'esclave (ID),</li> <li>• les sous-codes d'identification de l'esclave (ID1 et ID2).</li> </ul>
4	Adresse	Adresse physique de l'esclave.
<p><b>Remarque :</b> Les paramètres de fonctionnement, adresse, données de configuration et d'identification sont sauvegardés dans une mémoire non volatile.</p>		

## Principes de mise en œuvre logicielle

### Présentation

Pour respecter la philosophie adoptée dans TwidoSoft, l'utilisateur doit procéder par étapes pour créer une application AS-Interface.

### Principe de mise en œuvre

L'utilisateur doit savoir comment configurer de façon fonctionnelle son bus AS-Interface (Voir *Comment insérer un équipement esclave dans une configuration AS-Interface V2 existante*, p. 235).

Le tableau ci-dessous présente les différentes phases de mise en œuvre logicielle du bus AS-Interface V2.

Mode	Phase	Description
Local	Déclaration du coupleur	Choix de l'emplacement du module maître AS-Interface TWDNOI10M3 sur le bus d'expansion.
	Configuration de la voie du module	Choix des modes "maître".
	Déclaration des équipements esclaves	Choix pour chaque équipement : <ul style="list-style-type: none"> <li>de son numéro d'emplacement sur le bus,</li> <li>du type d'esclave adressage standard ou adressage étendu.</li> </ul>
	Validation des paramètres de configuration	Validation au niveau esclave.
	Validation globale de l'application	Validation de niveau application.
Local ou connecté	Symbolisation (optionnel)	Symbolisation des variables associées aux équipements esclaves.
	Programmation	Programmation de la fonction AS-Interface V2.
Connecté	Transfert	Transfert de l'application dans l'automate.
	Mise au point	Mise au point de l'application à l'aide : <ul style="list-style-type: none"> <li>de l'écran de mise au point permettant d'une part la visualisation des esclaves (adresse, paramètres), et d'autre part l'adressage des esclaves aux adresses souhaitées.</li> <li>des écrans de diagnostic permettant d'identifier les défauts.</li> </ul>

**Note** : La déclaration et la suppression du module maître AS-Interface sur le bus d'expansion se déroule comme pour un autre module d'expansion. Mais une fois deux modules maître AS-Interface déclarés sur le bus d'expansion, TwidoSoft ne permet plus d'en déclarer un autre.

---

**Précautions  
avant la  
connexion**

Avant de connecter (de façon logicielle) le PC à l'automate et pour éviter tout problème de détection :

- assurez-vous qu'il n'y a pas d'esclave présent physiquement sur le bus à l'adresse 0,
  - assurez-vous qu'il n'y a pas 2 esclaves présents physiquement à la même adresse.
-

## Description de l'écran de configuration du bus AS-Interface

**Présentation** L'écran de configuration du module maître AS-Interface donne accès aux paramètres associés au coupleur et aux équipements esclaves. Il permet la visualisation et la modification des paramètres en mode local.

**Illustration en mode local** Illustration de l'écran de configuration en mode local :

Configurer un module - TWDNOI10M3 [Position 1]

Description

Module d'expansion Maître AS-Interface

Configuration

Configuration AS-interface

Esclaves std /A

XVBC21A

WXA36

00  
01  
02  
03  
04  
05  
06  
07  
08  
09  
10  
11  
12  
13  
14  
15  
16

Esclaves /B

ASI20MT4IE

INOUT24/12

Esclave 1A

Caractéristiques

Profil : IO 7 ID f ID1 f ID2 f

Commentaire : Embase colonne lumineuse XVB

Paramètres

Bits

0 ☒ Clignotement e1

1 ☒ Clignotement e2

2 ☒ Clignotement e3

3 ☒ Clignotement e4

Entrées/Sorties

Entrées	Repère	Sorties	Repère
1	%IA1.1A.0	1	%QA1.1A.0
2	%IA1.1A.1	2	%QA1.1A.1

Mode maître

☒ Activation échange de données

☐ Arrêt réseau

☒ Adressage automatique

OK

Annuler

Aide



## Description de l'écran en mode local

Cet écran regroupe la totalité des informations constituant le bus en proposant trois blocs d'informations :

Blocs	Description
Configuration AS-interface	Image du bus souhaitée par l'utilisateur : visualisation des esclaves à adressage standard et étendu projetés (prévus) sur le bus. Il faut descendre le curseur de la barre verticale pour accéder aux adresses suivantes. Les adresses grisées correspondent à des adresses indisponibles pour y configurer un esclave. Si par exemple un nouvel esclave standard est déclaré à l'adresse 1A, l'adresse 1B est alors automatiquement grisée.
Esclave xxA/B	Configuration de l'esclave sélectionné : <ul style="list-style-type: none"> <li>● Caractéristiques : codes IO, ID, ID1 et ID2 (profiles), et commentaire sur l'esclave,</li> <li>● Paramètres : liste des paramètres (modifiables), sous forme binaire (4 cases à cocher) ou décimale (1 case) au choix de l'utilisateur,</li> <li>● Entrées/Sorties : liste des Entrées/Sorties disponibles, et leur repère (adresse).</li> </ul>
Mode maître	Activation ou désactivation possible des deux fonctionnalités disponibles pour ce coupleur AS-Interface (comme par exemple l'adressage automatique). La fonction "Arrêt du réseau" vous permet de forcer le bus AS-Interface pour entrer en mode local. Le mode "Adressage automatique" est coché par défaut. Remarque : La fonction "Activation échange de données" n'est pas encore disponible.

L' écran propose également 3 boutons :

Boutons	Description
OK	Permet de sauvegarder la configuration du bus AS-Interface visible à l'écran de configuration. Retour ensuite à l'écran principal. La configuration peut alors être transférée vers l'automate Twido.
Annuler	Retourne à l'écran principal sans prendre en compte les modifications en cours.
Aide	Ouvre une fenêtre d'aide à l'écran.

**Note** : Les modifications dans l'écran de configuration ne sont possibles qu'en mode local.

## Configuration du bus AS-Interface

---

### Introduction

La configuration du bus AS-Interface s'effectue dans l'écran de configuration en mode local.


Une fois le maître AS-Interface et les modes maître sélectionnés, la configuration du bus AS-Interface consiste à configurer les équipements esclaves.

---

**Procédure de  
déclaration et  
configuration  
d'un esclave**

Marche à suivre pour créer ou modifier un esclave sur le bus AS-Interface V2 :

Etape	Action
1	<p>Sur la cellule de l'adresse désirée (non grisée) dans l'image du bus :</p> <ul style="list-style-type: none"><li>Faites un double-clic : accès à l'étape 3</li></ul> <p>OU</p> <ul style="list-style-type: none"><li>Faites un clic droit de la souris :</li></ul> <p>Résultat :</p>
	<p>Remarque :</p> <p>Un menu contextuel apparaît. Il permet :</p> <ul style="list-style-type: none"><li>de configurer un nouvel esclave sur le bus,</li><li>de modifier la configuration de l'esclave désiré,</li><li>de copier (ou Ctrl+C), couper (ou Ctrl+X), coller un esclave (ou Ctrl+V),</li><li>de supprimer un esclave (ou Suppr).</li></ul>

Etape	Action
2	<p>Choisissez dans le menu contextuel :</p> <ul style="list-style-type: none"> <li>• "Nouveau" pour créer un nouvel esclave : un écran de configuration de l'esclave apparaît, le champ "Adresse" indique l'adresse sélectionnée, les champs de "Profil" sont à F par défaut et tous les autres champs de l'écran sont vides.</li> <li>• "Ouvrir" pour créer un nouvel esclave ou pour modifier la configuration de l'esclave sélectionné. Dans le cas d'un nouvel esclave, un nouvel écran pour configurer l'esclave apparaît, le champ "Adresse" indique l'adresse sélectionnée, les champs de "Profil" sont à F par défaut et tous les autres champs de l'écran sont vides. Dans le cas d'une modification, l'écran de configuration de l'esclave apparaît avec les champs contenant les valeurs préalablement définies de l'esclave sélectionné.</li> </ul> <p>Illustration d'un écran de configuration pour un nouvel esclave :</p> 
3	<p>Saisissez ou modifiez dans l'écran de configuration de l'esclave qui est alors affiché :</p> <ul style="list-style-type: none"> <li>• le nom du nouveau profil (limité à 13 caractères),</li> <li>• un commentaire (optionnel)</li> </ul> <p>ou cliquez sur le bouton "Catalogue..." et choisissez un esclave de la famille de profils AS-Interface pré-configuré.</p>

Etape	Action
4	<p>Saisissez :</p> <ul style="list-style-type: none"> <li>le code IO (correspond à la configuration entrée/sortie),</li> <li>le code ID (identificateur), plus ID1 et ID2 pour un type étendu.</li> </ul> <p>Remarque :</p> <p>Les champs "Entrées" et "Sorties" indiquent le nombre de voie d'entrée et de sortie. Ils sont implémentés automatiquement lors de la saisie du code IO.</p>
5	<p>Définissez pour chaque paramètre :</p> <ul style="list-style-type: none"> <li>sa prise en compte par le système (case cochée en vue "Bits", ou valeur décimale entre 0 et 15 en vue "Décimal"),</li> <li>un libellé plus significatif que "Paramètre X" (optionnel).</li> </ul> <p>Remarque :</p> <p>Les paramètres sélectionnés sont l'image des paramètres permanents à fournir au maître AS-Interface.</p>
6	<p>Modifiez "Adresse" si nécessaire (dans la limite des adresses disponibles sur le bus), en cliquant sur les flèches haut/bas à gauche de l'adresse (accès alors aux adresses autorisées) ou en saisissant directement l'adresse au clavier.</p>
7	<p>Validez la configuration de l'esclave en cliquant sur le bouton "OK".</p> <p>Le résultat est la vérification que :</p> <ul style="list-style-type: none"> <li>les codes IO et ID sont autorisés,</li> <li>l'adresse de l'esclave est autorisée (en cas de saisie clavier) selon le code ID (les esclaves "banque" /B sont seulement disponibles si le code ID est égal à A).</li> </ul> <p>En cas d'erreur, un message avertit l'utilisateur du type d'erreur (exemple : "L'esclave ne peut avoir cette adresse") et l'écran est réaffiché avec les valeurs initiales (dans profil ou adresse selon l'erreur).</p>

**Note :** Le logiciel limite le nombre de déclaration d'esclave analogique à 7.

**Note :** A propos du catalogue Schneider AS-Interface : lorsque vous cliquez sur le bouton Catalogue, "Vous pouvez créer et configurer des esclaves dans "Famille privée" (autre que ceux du catalogue Schneider AS-Interface.

## Catalogue AS-Interface

Le bouton catalogue sert à faciliter la configuration des esclaves sur le bus. Lorsque vous utilisez un esclave de la famille Schneider utilisez ce bouton, la configuration sera très simple et rapide

Lorsque vous cliquez sur le bouton "Catalogue" de la fenêtre "Configurer un esclave AS-Interface", vous ouvrez la fenêtre suivante :

**Catalogue AS-Interface**

Famille de profils AS-Interface:

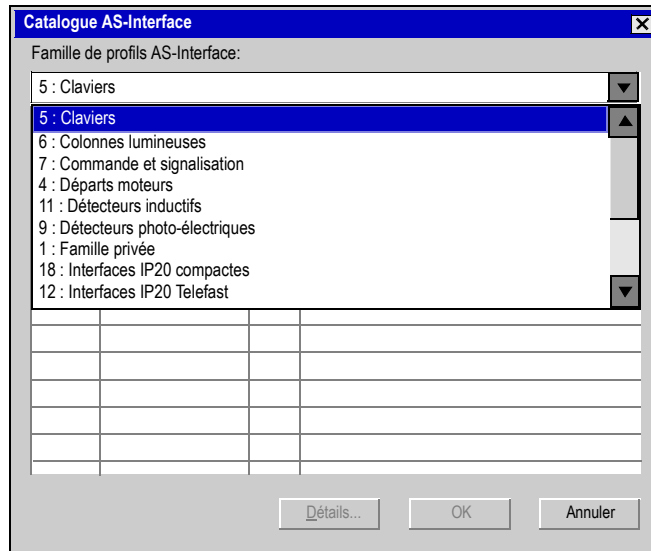
6 : Colonnes lumineuses

Catalogue AS-Interface: Colonnes lumineuses

Profil	Nom AS-Interface	@	Commentaire
7.F.F.F	XVBC21A	std	Embase colonne lumineuse XVB.
8.F.F.F	XVA-S102	std	Embase colonne lumineuse XVA.

Détails... OK Annuler

Dans le menu déroulant, vous avez accès à toutes les familles du catalogue AS-Interface Schneider :



Lorsque vous avez choisi votre famille, la liste des esclaves correspondant s'affiche. Cliquez sur l'esclave désiré et validez en cliquant sur "Ok"

**Note :** Vous pouvez afficher les caractéristiques d'un esclave en cliquant sur le bouton "Détails".

**Note :** Vous pouvez rajouter et configurer des esclaves qui ne font pas partie du catalogue Schneider. Il vous suffit de choisir la famille privée et de configurer ce nouvel esclave.

## Description de l'écran de mise au point

---

### Présentation

Quand le PC est **connecté** au contrôleur (après chargement de l'application vers l'automate), l'onglet de "Mise au point" apparaît à droite de celui de "Configuration", il permet l'accès à l'écran de mise au point.

L'écran de mise au point fournit, de façon dynamique, une image du bus physique incluant :

- la liste des esclaves prévus (saisis) pendant la configuration avec leur nom, et la liste des esclaves détectés (de nom inconnu si non prévus),
- l'état du coupleur AS-Interface et des équipements esclaves,
- l'image du profil, des paramètres et des valeurs des entrées/sorties des esclaves sélectionnés.

Il permet également à l'utilisateur :

- d'obtenir un diagnostic des esclaves en erreur (Voir *Visualisation des états des esclaves*, p. 226),
  - de modifier l'adresse d'un esclave en mode connecté (Voir *Modification de l'adresse d'un esclave*, p. 227),
  - de transmettre l'image des esclaves à l'écran de configuration (Voir *Mise à jour de la configuration du bus AS-Interface en mode connecté*, p. 229),
  - d'adresser tous les esclaves aux adresses souhaitées (lors de la première mise au point).
-



**Illustration de  
l'écran "Mise au  
point"**

L'illustration de l'écran de mise au point (en mode connecté uniquement) se présente ainsi :

Configurer un module - TWDNOI10M3 [Position 2]

Description

Module d'expansion Maître AS-Interface

Configuration

Mise au point

Configuration AS-interface V2

Esclaves std /A

00

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

XVBC21A

WXA36

Esclaves /B

ASI20MT4IE

INOUT24/12

Inconnu

Esclave 1A

Caractéristiques

Profil : IO 7 ID f ID1 f ID2 f

Commentaire : Embase colonne lumineuse XVB

Paramètres

Bits

Décimal

0

1

2

3

Entrées/Sorties

Entrées	Valeur	Format	Sorties	Valeur	Format
%IA1.1A.0	0	Déc	%QA1.1A.0	0	Déc
%IA1.1A.1	0	Déc	%QA1.1A.1	0	Déc

Erreur sur le réseau

Bus AS-Interface

Configuration OK	OFF	Adressage auto possible	OFF	Esclave adr 0 détecté	OFF	Coupure alim	OFF
Esclaves OK	ON	Mode protégé	OFF	Adressage auto actif	ON	Arrêt réseau	OFF

OK

Annuler

Aide

## Description de l'écran de mise au point

L'écran de "Mise au point" fournit les mêmes informations que l'écran de configuration (Voir *Description de l'écran en mode local*, p. 217). Les différences sont listées dans le tableau suivant :

Bloc	Description
Configuration AS-interface V2	Image du bus physique. Inclut l'état des esclaves : <ul style="list-style-type: none"> <li>● voyant vert : l'esclave à cette adresse est actif.</li> <li>● voyant rouge : l'esclave à cette adresse est en erreur, et un message informe du type d'erreur dans la fenêtre "Erreur sur le réseau".</li> </ul>
Esclave xxA/B	Image de la configuration de l'esclave sélectionné : <ul style="list-style-type: none"> <li>● Caractéristiques : image du profil détecté (grisées, non modifiable),</li> <li>● Paramètres : image des paramètres détectés. L'utilisateur peut uniquement choisir le format d'affichage des paramètres,</li> <li>● Entrées/Sorties : les valeurs des entrées/sorties détectées sont affichées, non modifiables.</li> </ul>
Erreur sur le réseau	Informe du type d'erreur si l'esclave sélectionné est en erreur.
Bus AS-Interface	Informations résultantes d'une commande implicite "Read Status". <ul style="list-style-type: none"> <li>● indique l'état du bus : par exemple "Configuration OK = OFF" indique que la configuration prévue par l'utilisateur ne correspond à la configuration physique du bus,</li> <li>● indique les fonctionnalités autorisées au module maître AS-Interface : par exemple "Adressage auto actif = ON" indique que le mode maître adressage automatique est autorisé.</li> </ul>

## Visualisation des états des esclaves

Lorsque le voyant associé à une adresse est rouge, l'esclave associé à cette adresse est en erreur. La fenêtre "Erreur sur le réseau" fournit alors le diagnostic de l'esclave sélectionné.

Descriptif des erreurs :

- le profil prévu par l'utilisateur en configuration à une adresse donnée, ne correspond pas au profil réel détecté à cette adresse sur le bus (diagnostic : "Erreur de profil"),
- un nouvel esclave non prévu en configuration, est détecté sur le bus : un voyant rouge est alors affiché pour cette adresse et le nom de l'esclave affiché est "Inconnu" (diagnostic : "Esclave non projeté"),
- défaut périphérique si l'esclave détecté le supporte (diagnostic : "Défaut périphérique"),
- un profil est prévu en configuration mais aucun esclave est détecté à cette adresse sur le bus (diagnostic : "Esclave non détecté").

## Modification de l'adresse d'un esclave

### Présentation

L'utilisateur peut, à travers l'écran de mise au point, modifier l'adresse d'un esclave en mode connecté.

### Modification de l'adresse d'un esclave

Le tableau suivant présente la procédure pour modifier l'adresse d'un esclave :

Etape	Désignation
1	Accédez à l'écran de "Mise au point".
2	Sélectionnez un esclave dans la zone "Configuration AS-interface V2"
3	Exécutez un "glisser et déposer" à l'aide de la souris vers la cellule correspondant à l'adresse désirée. illustration : glisser-déposer de l'esclave 3B vers l'adresse 15B

Configuration

Mise au point

Configuration AS-interface V2

Esclaves std /A

00

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

XVBC21A

WXA36

Esclaves /B

00

01

02

03

04

05

06

07

08

09

10

11

12

13

14

15

16

ASI20MT41E

INOUT24/12

ASI20MT41E

Inconnu

Etape	Désignation
Résultat :	Un contrôle de tous les paramètres de l'esclave s'effectue automatiquement pour vérifier si l'opération est possible.
illustration du résultat :	<div><div><div>Configuration</div><div>Mise au point</div></div><div><div>Configuration AS-interface V2</div><div><div><div>Esclaves std /A</div><div><div>00</div><div>01</div><div>02</div><div>03</div><div>04</div><div>05</div><div>06</div><div>07</div><div>08</div><div>09</div><div>10</div><div>11</div><div>12</div><div>13</div><div>14</div><div>15</div><div>16</div></div><div><div>XVBC21A</div><div></div><div></div><div></div><div></div><div></div><div>WXA36</div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div></div><div><div>Esclaves /B</div><div><div></div><div></div><div>ASI20MT41E</div><div></div><div>INOUT24/12</div><div></div><div></div><div></div><div></div><div></div><div>Inconnu</div><div></div><div></div><div>Inconnu</div><div></div></div></div></div></div></div>
Après l'opération, le diagnostic de l'esclave à l'adresse 3B affiche "esclave non détecté" indiquant que l'esclave prévu à cette adresse n'est plus présent. En sélectionnant l'adresse 15B, on retrouve bien le profil et les paramètres de l'esclave déplacé, le nom de l'esclave reste, quant à lui, inconnu car il n'était pas prévu à cette adresse là.	

**Note :** Le profil et les paramètres d'un esclave ne sont pas attachés à son nom. Plusieurs esclaves de noms différents peuvent avoir les mêmes profils et paramètres.

## Mise à jour de la configuration du bus AS-Interface en mode connecté

---

### Présentation

En mode connecté, aucune modification de l'écran de configuration n'est autorisée et la configuration physique et la configuration logicielle peuvent être différentes. Toute différence de profil ou paramètres d'un esclave prévu ou non en configuration peut être prise en compte dans l'écran de configuration, il est en effet possible de transmettre toute modification à l'écran de configuration avant de transférer la nouvelle application vers l'automate.

La procédure à suivre pour prendre en compte la configuration physique, est la suivante :

Etape	Désignation
1	Transfert de la configuration de l'esclave désiré vers l'écran de configuration.
2	Acceptation de la configuration dans l'écran de configuration.
3	Validation de la nouvelle configuration.
4	Transfert de l'application au coupleur.

---

Transfert de l'image d'un esclave vers la configuration

Dans le cas de la détection d'un esclave sur le bus non prévu dans la configuration, un esclave "Inconnu" apparaît dans la zone "Configuration AS-interface V2" de l'écran de mise au point à l'adresse détectée.  
Le tableau suivant indique la procédure pour transférer l'image de l'esclave "Inconnu" dans l'écran de configuration :

Etape	Désignation
1	Accédez à l'écran de "Mise au point"
2	Sélectionnez l'esclave désiré dans la zone "Configuration AS-interface V2".
3	Exécutez un clic droit sur la souris pour choisir "Transfert Conf". illustration : <div data-bbox="577 496 1115 1183"></div>
	Résultat : L'image de l'esclave sélectionné (image du profil et paramètres) est alors transféré à l'écran de configuration.
4	Recommencez l'opération pour chacun des esclaves dont on veut transférer l'image vers l'écran de configuration.

## Retour à l'écran de configuration

Quand l'utilisateur revient dans l'écran de configuration, tous les nouveaux esclaves (non prévus) transférés sont visibles.

Illustration de l'écran de configuration après le transfert de tous les esclaves :

Configuration AS-interface V2	
Esclaves std /A	Esclaves /B
XVBC21A	
	ASI20MT4IE
	X INOUT24/12
WXA36	
	! Inconnu
	! Inconnu

Légende :

- la croix signifie qu'il y a des différences entre l'image du profil de l'esclave transféré, et le profil souhaité initialement dans l'écran de configuration.
- le point d'exclamation signifie qu'un nouveau profil a été introduit dans l'écran de configuration.

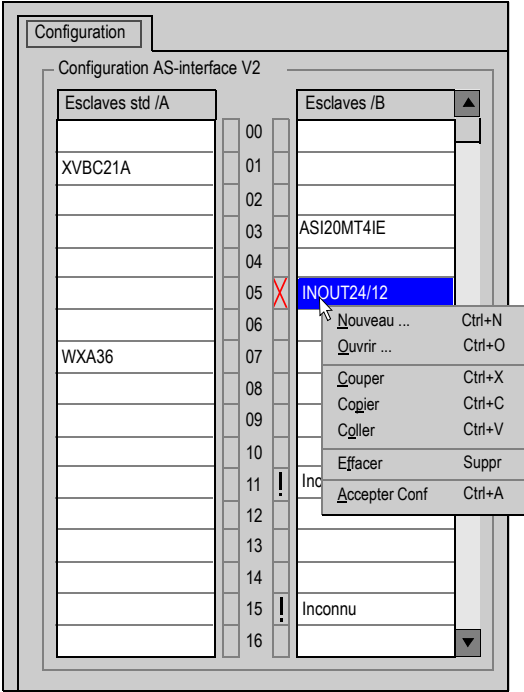
Explication :

L'écran de configuration présente toujours l'image permanente de la configuration souhaitée (d'où la présence de l'esclave en 3B malgré le changement d'adresse (Voir *Modification de l'adresse d'un esclave*, p. 227)), complétée de l'image courante du bus.

Les profils et paramètres des esclaves prévus qui sont affichés correspondent à ceux qui étaient prévus. Les profils et paramètres des esclaves inconnus qui sont affichés correspondent aux images de ceux détectés.

Procédure pour le transfert de l'application définitive vers le coupleur

Avant de transférer une nouvelle application vers le coupleur, l'utilisateur peut pour chacun des esclaves prendre en compte l'image du profil et paramètres détectée (transférée à l'écran de configuration) ou modifier la configuration "à la main" (Voir *Procédure de déclaration et configuration d'un esclave*, p. 219).  
Le tableau suivant décrit la marche à suivre pour la validation et le transfert de la configuration définitive vers le coupleur :

Etape	Action
1	Déconnectez de façon logicielle le PC du coupleur. remarque : Aucune modification dans l'écran de configuration n'est possible si le PC est connecté au coupleur.
2	Faites un clic droit souris sur l'esclave désiré.
3	2 choix : <ul style="list-style-type: none"><li>choisissez "Accepter Conf" pour accepter le profil <b>détecté</b> de l'esclave sélectionné.</li></ul> illustration : <div></div> <p>Pour chacun des esclaves marqués d'une croix, un message avertit l'utilisateur que cette opération écrasera le profil initial (affiché dans l'écran) de l'esclave.</p> <ul style="list-style-type: none"><li>choisissez les autres choix du menu contextuel pour configurer à la main l'esclave sélectionné.</li></ul>



<b>Etape</b>	<b>Action</b>
4	Recommencez l'opération pour chacun des esclaves désirés dans la configuration.
5	Pressez sur le bouton "OK" pour valider et créer la nouvelle application. Résultat : retour automatique à l'écran principal.
6	Transférez l'application vers le coupleur.

---

## Adressage automatique d'un esclave AS-Interface V2

---

### Présentation

Chaque esclave présent sur le bus AS-Interface doit se voir affecter (par configuration) une adresse physique unique. Celle-ci doit être l'image de celle déclarée dans TwidoSoft.

Le logiciel TwidoSoft offre un service d'adressage automatique des esclaves qui évite ainsi d'utiliser une console AS-Interface.

Le service d'adressage automatique est utilisé pour :

- remplacer un esclave défaillant,
- insérer un nouvel esclave.

### Marche à suivre

Le tableau ci-dessous présente la marche à suivre pour définir le paramètre **Adressage automatique**.

Etape	Action
1	Accédez à l'écran de configuration du module maître AS-Interface V2.
2	<p>Cliquez sur la case à cocher <b>Adressage automatique</b> située dans la zone <b>Mode maître</b>.</p> <p><b>Résultat</b> : Le service <b>Adressage automatique</b> sera actif (case cochée) ou non actif (case non cochée).</p> <p><b>Remarque</b> : Le paramètre <b>Adressage automatique</b> est sélectionné par défaut dans l'écran de configuration.</p>

## Comment insérer un équipement esclave dans une configuration AS-Interface V2 existante

### Présentation

Il est possible d'insérer un équipement dans une configuration AS-Interface V2 existante sans avoir recours à l'utilisation du programmeur de poche.

Cette opération est possible dès lors que :

- le service **Adressage automatique** du mode de configuration est actif (Voir *Adressage automatique d'un esclave AS-Interface V2, p. 234*),
- un seul esclave est absent dans la configuration physique,
- l'esclave à insérer est prévu dans l'écran de configuration,
- l'esclave possède le profil attendu par la configuration,
- l'esclave possède l'adresse 0(A).

Ainsi, le coupleur AS-Interface V2 affectera automatiquement à l'esclave la valeur prédéfinie dans la configuration.

### Marche à suivre

Le tableau ci-dessous présente la marche à suivre pour que l'insertion automatique d'un nouvel esclave soit effective.

Etape	Action
1	Ajoutez le nouvel esclave dans l'écran de configuration en mode local.
2	Faites un transfert de configuration vers l'automate en mode connecté.
3	Raccordez physiquement le nouvel esclave d'adresse 0(A) sur le bus AS-Interface V2.

**Note :** Il est possible de modifier une application en réalisant la manipulation ci-dessus autant de fois que nécessaire.

## Remplacement automatique d'un esclave AS-Interface V2 défectueux

---

### Principe

Lorsqu'un esclave est déclaré en défaut, il est possible de le remplacer de façon automatique par un esclave de même type.

Le remplacement s'effectue sans arrêt du bus AS-Interface V2 et sans manipulation particulière dès lors que le service **Adressage automatique** du mode de configuration est actif (Voir *Adressage automatique d'un esclave AS-Interface V2*, p. 234).

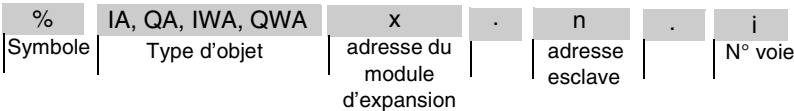
Deux possibilités peuvent se présenter :

- l'esclave venant en remplacement est programmé avec la même adresse à l'aide du programmeur de poche et possède le même profil et sous-profil que l'esclave défectueux. Il sera donc inséré automatiquement dans la liste des esclaves détectés (LDS) et dans la liste des esclaves actifs (LAS),
  - l'esclave venant en remplacement est vierge (adresse 0(A), esclave neuf) et possède le même profil que l'esclave défectueux. Il prendra automatiquement l'adresse de l'esclave remplacé et sera donc inséré dans la liste des esclaves détectés (LDS) et dans la liste des esclaves actifs (LAS).
-

# Adressage des entrées/sorties associées aux équipements esclaves connectés sur bus AS-Interface V2

**Présentation** Cette page présente les spécificités de l'adressage des entrées/sorties TOR ou analogiques des équipements esclaves. Pour éviter toute confusion avec les E/S déportés, de nouveaux symboles sont proposés avec une syntaxe AS-Interface : %**I**A par exemple.

**Illustration** Rappel du principe d'adressage :



**Valeurs spécifiques** Le tableau ci-dessous donne les valeurs spécifiques aux objets des esclaves AS-Interface V2 :

Elément	Valeurs	Commentaire
IA	-	Image de l'entrée physique TOR de l'esclave.
QA	-	Image de la sortie physique TOR de l'esclave.
IWA	-	Image de l'entrée physique analogique de l'esclave.
QWA	-	Image de la sortie physique analogique de l'esclave.
x	1 à 7	Adresse du module AS-Interface sur le bus d'expansion
n	0A à 31B	L'emplacement 0 n'est pas configurable.
i	0 à 3	-

**Exemples**

Le tableau présente quelques exemples d’adressage des E/S :

Objet d’E/S	Description
%IWA4.1A.0	entrée analogique 0 de l’esclave1A du module AS-Interface positionné en 4 sur le bus d’expansion.
%QA2.5B.1	sortie TOR 1 de l’esclave 5B du module AS-Interface positionné en 2 sur le bus d’expansion.
%IA1.12A.2	entrée TOR 2 de l’esclave 12A du module AS-Interface positionné en 1 sur le bus d’expansion.

**Echanges implicites**

Les objets décrits ci-dessus sont échangés de façon implicite, c’est à dire qu’ils sont échangés de façon automatique à chaque cycle automate.

---

# Programmation et diagnostic du bus AS-Interface V2

## Echanges explicites

Des objets (mots et bits) associés au bus AS-Interface apportent des informations (ex : fonctionnement du bus, état des esclaves...) et des commandes supplémentaires pour effectuer une programmation avancée de la fonction AS-Interface. Ces objets sont échangés de façon explicite entre l'automate Twido et le maître AS-Interface par le bus d'expansion :

- sur demande du programme utilisateur à l'aide de l'instruction : ASI\_CMD (voir plus bas "Présentation de l'instruction ASI\_CMD"),
- via l'écran de mise au point ou la table d'animation.

## Mots systèmes spécifiques réservés

Les mots systèmes réservés dans l'automate Twido pour les modules maîtres AS-Interface permettent de connaître l'état du réseau : %SW73 est réservé pour le premier module d'expansion AS-Interface, et %SW74 pour le second. Seules les 5 premiers bits de ces mots sont utilisés, ils sont en lecture seule.

Le tableau suivant présente les bits utilisés :

Mots système	Bit	Description
%SW73 et %SW74	0	état du système ( = 1 si configuration OK, 0 sinon)
	1	échange de données ( = 1 si échange de données activée, 0 si en mode Data Exchange Off (Voir <i>Mode de fonctionnement du module interface bus AS-Interface V2</i> , p. 244))
	2	système en stop ( = 1 si le mode Offline (Voir <i>Mode Offline</i> , p. 244) est activé, 0 sinon)
	3	instruction ASI_CMD terminée ( = 1 si terminée, 0 si en cours)
	4	erreur instruction ASI_CMD ( = 1 si erreur dans instruction, 0 sinon)

Exemple d'utilisation (pour le premier module d'expansion AS-Interface):

Avant d'utiliser une instruction ASI\_CMD, le bit %SW73:X3 doit être vérifié pour savoir si une instruction n'est pas en cours : vérifier que %SW73:X3 = 1.

Pour savoir, si l'instruction s'est ensuite bien exécutée, vérifier que le bit %SW73:X4 est égal à 0.

## Présentation de l'instruction ASI\_CMD

Par programme utilisateur, l'instruction ASI\_CMD permet à l'utilisateur de programmer son réseau et d'obtenir le diagnostic des esclaves. Les paramètres de l'instruction sont passés par mots internes (mémoires) %MWx.

La syntaxe de l'instruction est la suivante :

**ASI\_CMD<sub>n</sub> %MW<sub>x</sub>:l**

légende :

Symbole	Désignation
n	adresse du module d'expansion AS-Interface (1 à 7).
x	numéro du premier mot interne (mémoire) passé en paramètre (0 à 254).
l	longueur de l'instruction en nombre de word (2).

## Utilisation de l'instruction ASI\_CMD

Le tableau suivant décrit l'action de l'instruction ASI\_CMD en fonction de la valeur des paramètres %MW(x), et %MW(x+1) quand nécessaire. Pour les demandes de diagnostic des esclaves, le résultat est retourné dans %MW(x+1).

%MW <sub>x</sub>	%MW <sub>x+1</sub>	Action
1	0	quitte le mode Offline.
1	1	passse en mode Offline.
2	0	interdit l'échange de données entre le maître et ses esclaves (entre dans le mode Data Exchange Off).
2	1	autorise l'échange de données entre le maître et ses esclaves (sort du mode Data Exchange Off).
3	réservé	-
4	Résultat	lit la liste des esclaves actifs (table LAS) de l'adresse 0A à 15A (1 bit par esclave).
5	Résultat	lit la liste des esclaves actifs (table LAS) de l'adresse 16A à 31A (1 bit par esclave).
6	Résultat	lit la liste des esclaves actifs (table LAS) de l'adresse 0B à 15B (1 bit par esclave).
7	Résultat	lit la liste des esclaves actifs (table LAS) de l'adresse 16B à 31B (1 bit par esclave).
8	Résultat	lit la liste des esclaves détectés (table LDS) de l'adresse 0A à 15A (1 bit par esclave).
9	Résultat	lit la liste des esclaves détectés (table LDS) de l'adresse 16A à 31A (1 bit par esclave).
10	Résultat	lit la liste des esclaves détectés (table LDS) de l'adresse 0B à 15B (1 bit par esclave).
11	Résultat	lit la liste des esclaves détectés (table LDS) de l'adresse 16B à 31B (1 bit par esclave).



%MWx	%MWx+1	Action
12	Résultat	lit la liste des défauts périphériques des esclaves (table LPF) de l'adresse 0A à 15A (1 bit par esclave).
13	Résultat	lit la liste des défauts périphériques des esclaves (table LPF) de l'adresse 16A à 31A (1 bit par esclave).
14	Résultat	lit la liste des défauts périphériques des esclaves (table LPF) de l'adresse 0B à 15B (1 bit par esclave).
15	Résultat	lit la liste des défauts périphériques des esclaves (table LPF) de l'adresse 16B à 31B (1 bit par esclave).
16	Résultat	lit l'état du bus. Voir détail du résultat dans le paragraphe suivant.

**Note :** L'état du bus est mis à jour à chaque cycle automate. Mais le résultat de l'instruction ASI\_CMD de lecture du bus n'est disponible qu'au cycle automate suivant.

**Détail du résultat  
de l'instruction  
ASI\_CMD pour  
lire l'état du bus**

Dans le cas d'une lecture de l'état du bus par l'instruction ASI\_CMD (valeur du paramètre %MWx égale à 16), le format du résultat dans le mot %MWx+1 est la suivante :

%MWx+1		Désignation (1=OK, 0=NOK)
poids faible	bit 0	Configuration OK
	bit 1	LDS.0 (esclave présent à l'adresse 0)
	bit 2	Auto adressage actif
	bit 3	Auto adressage disponible
	bit 4	Mode Configuration actif
	bit 5	Normal opération active
	bit 6	APF (problème d'alimentation)
	bit 7	Offline prêt
poids fort	bit 0	Défaut périphérique
	bit 1	Echange de données actif
	bit 2	Mode Offline
	bit 3	Mode normal (1)
	bit 4	Défaut communication avec le maître AS-Interface
	bit 5	Instruction ASI_CMD en cours
	bit 6	Instruction ASI_CMD en erreur

**Détail du résultat de l'instruction ASI\_CMD pour lire l'état des esclaves**

Dans le cas d'un diagnostic des esclaves par l'instruction ASI\_CMD (valeur %MWx comprise entre 4 et 15), l'état des esclaves est retourné dans les bits (1=OK) du mot %MWx+1. Le tableau suivant donne le détail du résultat en fonction de la valeur du mot %MWx :

%MWx	%MWx+1															
valeur	octet poids fort								octet poids faible							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
4, 8, 12	15A	14A	13A	12A	11A	10A	9A	8A	7A	6A	5A	4A	3A	2A	1A	0A
5, 9, 13	31A	30A	29A	28A	27A	26A	25A	24A	23A	22A	21A	20A	19A	18A	17A	16A
6, 10, 14	15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	0B
7, 11, 15	31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B

Pour lire si l'esclave 20B est actif, l'instruction ASI\_CMD doit être exécutée avec le mot interne %MWx de valeur 7. Le résultat est retourné dans le mot interne %MWx+1, l'état de l'esclave 20B est donné par la valeur du bit 4 de l'octet de poids faible : si le bit 4 est égal à 1 alors l'esclave 20B est actif.

**Exemples de programmation de l'instruction ASI\_CMD**

Pour forcer le passage du maître AS-Interface (positionné en 1 sur le bus d'expansion) en mode Offline :

```
LD 1
[%MW0 := 16#0001]
[%MW1 := 16#0001]
LD %SW73:X3 //Si aucune instruction ASI_CMD est en cours, on continue
[ASI_CMD1 %MW0:2] //pour forcer le passage en mode Offline
```

Pour lire la table des esclaves actifs de l'adresse 0A à 15A :

```
LD 1
[%MW0 := 16#0004]
[%MW1 := 16#0000 //optionnel]
LD %SW73:X3 //Si aucune instruction ASI_CMD est en cours, on continue
[ASI_CMD1 %MW0:2] //pour lire la table LAS de l'adresse 0A à 15A
```

## Mode de fonctionnement du module interface bus AS-Interface V2

---

### Présentation

Le module interface bus AS-Interface TWDNOI10M3 dispose de trois modes de fonctionnement répondant chacun à des besoins particuliers. Ces modes sont :

- le mode protégé,
- le mode Offline
- le mode Data Exchange Off.

L'utilisation de l'instruction `ASI_CMD` (Voir *Présentation de l'instruction ASI\_CMD*, p. 240) dans un programme utilisateur permet de rentrer ou de sortir de ces modes.

---

### Mode protégé

Le mode de fonctionnement protégé est le mode généralement utilisé pour une application en exploitation. Il implique que le coupleur AS-Interface V2 soit configuré dans TwidoSoft. Celui-ci :

- vérifie en permanence que la liste des esclaves détectés est égale à la liste des esclaves prévus,
- surveille l'alimentation.

Dans ce mode, un esclave ne sera activé que s'il a été déclaré dans la configuration et détecté.

A la mise sous tension ou pendant la phase de configuration, l'automate Twido force le module AS-Interface en mode protégé.

---

### Mode Offline

A l'arrivée dans le mode Offline, le coupleur effectue d'abord une remise à zéro de tous les esclaves présents et arrête les échanges sur le bus. Pendant le mode Offline, les sorties sont forcées à zéro.

En dehors de l'usage du bouton PB2 sur le module AS-Interface TWDNOI10M3, le mode Offline est accessible de façon logicielle par l'instruction `ASI_CMD` (Voir *Exemples de programmation de l'instruction ASI\_CMD*, p. 243), de même pour quitter le mode et revenir au mode protégé.

---

### Mode Data Exchange Off

A l'arrivée dans le mode Data Exchange Off, les échanges sur le bus continuent à fonctionner, mais les données ne sont plus rafraîchies.

Ce mode n'est accessible que par l'instruction `ASI_CMD` (Voir *Utilisation de l'instruction ASI\_CMD*, p. 240).

---

---

# Fonctionnement de l'afficheur

# 10

---

## Présentation

### Objet de ce chapitre

Ce chapitre offre des informations sur l'utilisation de l'afficheur optionnel Twido.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Afficheur	246
Informations d'identification et états de l'automate	249
Variables et objets système	251
Paramètres de port série	258
Horloge calendaire	259
Facteur de correction de l'horodateur	260

## Afficheur

---

### Introduction

L'afficheur est une option de Twido qui permet d'afficher et de contrôler les données de l'application et quelques fonctions de l'automate, telles que l'état de fonctionnement et l'horodateur (RTC). Cette option est disponible sous la forme d'une cartouche (TWDXCPODC) pour les automates compacts ou d'un module d'expansion (TWDXCPODM) pour les automates modulaires.

L'afficheur dispose de deux modes de fonctionnement :

- Mode affichage : affiche simplement les données.
- Mode édition : permet de modifier les données.

**Note :** L'afficheur est mis à jour selon un intervalle défini dans le cycle de scrutation de l'automate. Cela peut provoquer des erreurs d'interprétation de l'affichage des sorties dédiées pour les impulsions %PLS et %PWM. Au moment où ces sorties sont échantillonnées, leur valeur est toujours égale à zéro et est affichée.

### Ecrans et fonctions

L'afficheur propose différents écrans à partir desquels vous pouvez accéder aux fonctions associées.

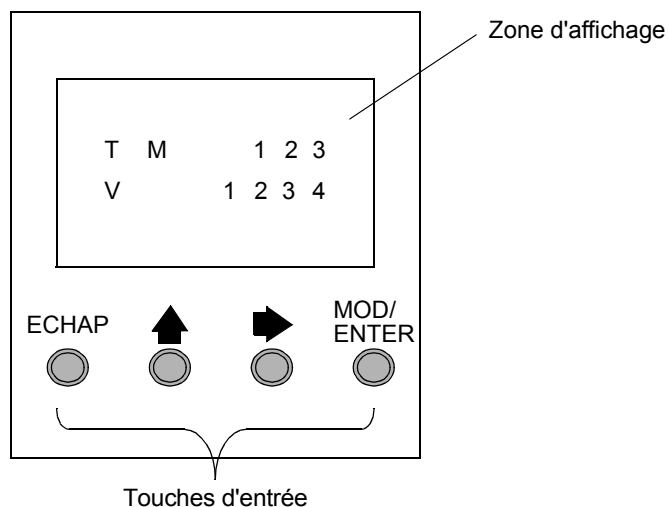
- Informations sur l'identification et l'état de l'automate : écran Opérations  
Affiche la révision du microprogramme et l'état de l'automate. Modifie l'état de l'automate à l'aide des commandes d'exécution, d'initialisation et d'arrêt.
- Variables et objets système : écran Données  
Sélection des données de l'application par le repère : %I, %Q et tous les autres objets logiciels de la base automate. Contrôle et modification de la valeur de l'objet donnée logiciel sélectionné.
- Paramètres du port série : écran Communications  
Affichage et modification des paramètres du port de communication.
- Horloge calendaire : écran Date/Heure  
Affichage et configuration de la date et de l'heure courantes (lorsque l'horodateur est installé).
- Correction de l'horodateur : facteur RTC  
Affichage et modification de la valeur de correction de l'horodateur en option.

**Note :**

1. Les automates compacts TWDLCA•40DRF disposent d'un horodateur intégré.
2. Pour tous les autres automates, l'horloge calendaire et la correction de l'horodateur ne sont disponibles que lorsque la cartouche horodateur en option (TWDXCPRTC) est installée.

**Illustration**

L'illustration suivante présente une vue de l'afficheur. Il est composé d'une zone d'affichage et de quatre touches d'entrée.

**Zone d'affichage**

L'afficheur est composé d'un écran à cristaux liquides pouvant afficher jusqu'à deux lignes de caractères.


- La première ligne de l'écran est composée de trois caractères de 13 segments et de quatre caractères de 7 segments.
- La seconde ligne est composée d'un caractère de 13 segments, d'un caractère de 3 segments (pour les signes plus et moins) et de cinq caractères de 7 segments.

**Touches d'entrée**


Les fonctions des quatre touches d'entrée dépendent du mode de l'afficheur.

Touche	En mode affichage	En mode édition
ECHAP		Annulation des modifications et retour à l'écran précédent.
▲		Accès à la valeur suivante d'un objet en cours de modification.
▶	Passage à l'écran suivant.	Accès au type d'objet suivant à modifier.
MOD/ ENTER	Passage en mode édition.	Validation des modifications et retour à l'écran précédent.

**Sélection et  
navigation entre  
les écrans**

L'affichage ou l'écran initial de l'afficheur présente des informations sur l'identification et l'état de l'automate. Appuyez sur la touche  pour passer d'un écran à l'autre. Les écrans de l'horloge calendaire ou le facteur de correction RTC apparaissent uniquement lorsque la cartouche horodateur en option (TWDXCPRTC) est détectée sur l'automate.

Appuyez sur la touche ECHAP pour revenir à l'écran initial. Dans la plupart des écrans, le fait d'appuyer sur la touche ECHAP permet de revenir à l'écran relatif aux informations d'identification et d'état de l'automate. Le fait d'appuyer sur la touche ECHAP permet de revenir à la saisie du premier objet système ou de l'objet système initial uniquement lors de la modification de variables et d'objets système autres que l'entrée initiale (%I0.0.0).

Pour modifier la valeur d'un objet, appuyez à nouveau sur la touche MOD/ENTER au lieu d'appuyer sur la touche  pour accéder au premier chiffre de la valeur.

---



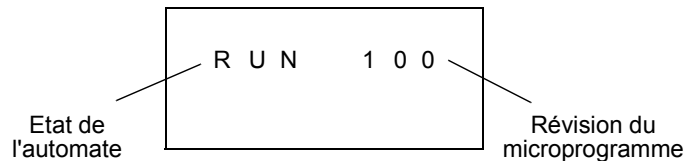
## Informations d'identification et états de l'automate

### Introduction

L'écran initial de l'afficheur optionnel Twido présente des informations sur l'identification et sur l'état de l'automate.

### Exemple

Comme l'illustre le schéma suivant, la version du microprogramme est affichée dans le coin supérieur droit de la zone d'affichage, l'état de l'automate dans le coin supérieur gauche.



### Etats de l'automate

L'automate peut se trouver dans l'un des états suivants :

- **NCF : Non configuré**

L'automate demeure en état NCF jusqu'à ce qu'une application soit chargée.

Aucun autre état n'est permis avant le chargement du programme de l'application. Vous pouvez tester les E/S en modifiant le bit système S8 (reportez-vous à la rubrique *Bits système (%S)*, p. 546).

- **STP : Arrêté**

Dès qu'une application est chargée sur l'automate, ce dernier passe à l'état STP. Dans cet état, l'application ne fonctionne pas. Les entrées sont mises à jour et les valeurs des données restent inchangées. Les sorties ne sont pas mises à jour dans cet état.

- **INI : Initial**

Seul un automate se trouvant à l'état STP peut passer à l'état INI. L'application n'est pas en cours d'exécution. Les entrées de l'automate sont mises à jour et les valeurs des données retrouvent leur état initial. Aucune sortie n'est mise à jour dans cet état.

- **RUN : En cours d'exécution**

Dans cet état, l'application fonctionne. Les entrées de l'automate sont mises à jour et les valeurs des données sont réglées par l'application. Il s'agit du seul état au cours duquel les sorties sont mises à jour.

- **HLT : Halt (Erreur d'application utilisateur)**

L'exécution de l'application est arrêtée dès que l'automate passe à l'état ERR.

Les entrées sont mises à jour et les valeurs des données restent inchangées.

Dans cet état, les sorties ne sont pas mises à jour. Dans ce mode, le code de l'erreur est affiché dans la partie inférieure droite de l'afficheur. Ce code prend la forme d'une valeur décimale sans signe.



- **NEX : Not Executable (non exécutable)**

Une modification en ligne a été apportée à la logique utilisateur. Conséquence : l'application n'est plus exécutable. Elle ne retrouvera cet état qu'une fois que toutes les causes de l'état Non Exec auront été résolues.

---

**Affichage et  
modification des  
états de  
l'automate**

L'afficheur vous permet de faire passer l'automate de l'état STP à l'état INI, de l'état STP à l'état RUN, ou de l'état RUN à l'état STP. Pour modifier l'état de l'automate, procédez comme suit :

Etape	Action
1	Appuyez sur la touche  jusqu'à ce que l'écran Affichage des états de l'automate apparaisse (ou appuyez sur la touche ESC). L'état courant de l'automate apparaît dans le coin supérieur gauche de la zone d'affichage.
2	Appuyez sur la touche MOD/ENTER pour passer en mode édition.
3	Appuyez sur la touche  pour sélectionner un état de l'automate.
4	Appuyez sur la touche MOD/ENTER pour accepter la valeur modifiée, ou sur la touche ESC pour ignorer les modifications apportées en mode édition.

---

## Variables et objets système

### Introduction

L'afficheur optionnel permet de contrôler et d'ajuster les données de l'application à l'aide des fonctionnalités suivantes :

- sélection des données de l'application par le repère (%I ou %Q, par exemple) ;
- contrôle de la valeur de l'objet/variable logiciel(le) sélectionné(e) ;
- modification de la valeur de l'objet donnée actuellement affiché (y compris le forçage des entrées et des sorties).

### Variables et objets système

Le tableau suivant répertorie, dans leur ordre d'accès, les variables et objets système qui peuvent être affichés et modifiés via l'afficheur.

Objet	Variable/Attribut	Description	Accès
Entrée	%Ix.y.z	Valeur	Lecture/Forçage
Sortie	%Qx.y.z	Valeur	Lecture/Ecriture/ Forçage
Temporisateur	%TMX.V %TMX.P %TMX.Q	Valeur courante Valeur de présélection Terminé	Lecture/Ecriture Lecture/Ecriture Lecture
Compteur	%Cx.V %Cx.P %Cx.D %Cx.E %Cx.F	Valeur courante Valeur de présélection Terminé Vide Plein	Lecture/Ecriture Lecture/Ecriture Lecture Lecture Lecture
Bit mémoire	%Mx	Valeur	Lecture/Ecriture
Mémoire mot	%MWx	Valeur	Lecture/Ecriture
Mot constante	%KWx	Valeur	Lecture
Bit système	%Sx	Valeur	Lecture/Ecriture
Mot système	%SWx	Valeur	Lecture/Ecriture
Entrée analogique	%IWx.y.z	Valeur	Lecture
Sortie analogique	%QWx.y.z	Valeur	Lecture/Ecriture
Compteur rapide (FC)	%FCx.V %FCx.VD* %FCx.P %FCx.PD* %FCx.D	Valeur courante Valeur courante Valeur de présélection Valeur de présélection Terminé	Lecture Lecture Lecture/Ecriture Lecture/Ecriture Lecture

Objet	Variable/Attribut	Description	Accès
Compteur rapide (VFC)	%VFCx.V %VFCx.VD* %VFCx.P %VFCx.PD* %VFCx.U %VFCx.C %VFCx.CD* %VFCx.S0 %VFCx.S0D* %VFCx.S1 %VFCx.S1D* %VFCx.F %VFCx.T %VFCx.R %VFCx.S	Valeur courante Valeur courante Valeur de présélection Valeur de présélection Sens de comptage Valeur de capture Valeur de capture Valeur de seuil 0 Valeur seuil 0 Valeur de seuil 1 Valeur de seuil 1 Sortie pour Base temps Activation sortie réflexe Activation entrée réflexe	Lecture Lecture Lecture/Ecriture Lecture/Ecriture Lecture Lecture Lecture Lecture/Ecriture Lecture/Ecriture Lecture/Ecriture Lecture/Ecriture Lecture/Ecriture Lecture/Ecriture Lecture/Ecriture
Mot entrée réseau	%INWx.z	Valeur	Lecture
Mot sortie réseau	%QNWx.z	Valeur	Lecture/Ecriture
Grafcet	%Xx	Bit étape	Lecture
Générateur d'impulsions	%PLS.N %PLS.ND* %PLS.P %PLS.D %PLS.Q	Nombre d'impulsions Nombre d'impulsions Valeur de présélection Terminé Sortie courante	Lecture/Ecriture Lecture/Ecriture Lecture/Ecriture Lecture Lecture
Modulateur de largeur d'impulsion	%PWM.R %PWM.P	Rapport Valeur de présélection	Lecture/Ecriture Lecture/Ecriture
Programmateur cyclique	%DRx.S %DRx.F	Numéro du pas courant Plein	Lecture Lecture
Fonction pas à pas	%SCx.n	Bit de fonction pas à pas	Lecture/Ecriture
Registre	%Rx.I %Rx.O %Rx.E %Rx.F	Entrée Sortie Vide Plein	Lecture/Ecriture Lecture/Ecriture Lecture Lecture
Registre bits à décalage	%SBR.x.yy	Bit de registre	Lecture/Ecriture
Message	%MSGx.D %MSGx.E	Terminé Erreur	Lecture Lecture
Entrée esclave AS-Interface	%IAx.y.z	Valeur	Lecture/Forçage







Objet	Variable/Attribut	Description	Accès
Entrée analogique esclave AS-Interface	%IWAx.y.z	Valeur	Lecture
Sortie esclave AS-Interface	%QAx.y.z	Valeur	Lecture/Ecriture/ Forçage
Sortie analogique esclave AS-Interface	%QWAx.y.z	Valeur	Lecture/Ecriture

Remarques :

1. (\*) correspond à une variable de mot double 32 bits. L'option de mot double est disponible sur tous les automates à l'exception des automates Twido TWDLC•A10DRF.
2. Etant donné que Twido utilise l'affectation de mémoire dynamique, les variables n'apparaîtront pas si elles ne sont pas utilisées dans une application.
3. Si la valeur de %MW est supérieure à +32 767 ou inférieure à -32 768, l'afficheur continue de clignoter.
4. Si la valeur de %SW est supérieure à 65 535, l'afficheur continue de clignoter, sauf pour %SW0 et pour %SW11. Lorsqu'une valeur dépassant les limites est entrée, elle est remplacée par la valeur configurée.
5. Lorsqu'une valeur dépassant les limites est entrée pour %PLS.P, la valeur écrite est la valeur de saturation.

### Affichage et modification des objets et des variables

Vous pouvez accéder à chaque type d'objet système en commençant par l'objet entrée (%I), en progressant de façon séquentielle jusqu'à l'objet message (%MSG) et en revenant finalement à l'objet entrée (%I).  
Pour afficher un objet système, procédez comme suit :

Etape	Action
1	Appuyez sur la touche  jusqu'à ce que l'écran Affichage des données apparaisse. L'objet Entrée ("I") apparaît dans le coin supérieur gauche de la zone d'affichage. La lettre "I" (ou le nom de l'objet précédemment visualisé en donnée) ne clignote pas.
2	Appuyez sur la touche MOD/ENTER pour passer en mode édition. La lettre "I" de l'objet Entrée (ou le nom de l'objet précédemment visualisé en donnée) commence à clignoter.
3	Appuyez sur la touche  pour progresser de façon séquentielle dans la liste des objets.
4	Appuyez sur la touche  pour progresser de façon séquentielle dans le champ d'un type d'objet et appuyez sur la touche  pour incrémenter la valeur de ce champ. Utilisez les touches  et  pour consulter et modifier tous les champs de l'objet affiché.
5	Répétez les étapes 3 et 4 jusqu'à ce que l'édition soit terminée.
6	Appuyez sur la touche MOD/ENTER pour accepter les valeurs modifiées. Remarque : Le nom et le repère de l'objet doivent être validés pour pouvoir accepter ces modifications. Cela signifie qu'ils doivent exister dans la configuration de l'automate avant de pouvoir utiliser l'afficheur. Appuyez sur la touche ECHAP pour annuler les modifications apportées en mode édition.

### Valeurs des données et formats d'affichage

En général, la valeur des données pour un objet ou une variable est affichée comme un entier avec signe ou sans signe dans la partie inférieure droite de la zone d'affichage. Les zéros non significatifs sont supprimés de tous les champs pour l'affichage des valeurs. Le repère de chaque objet apparaît dans l'afficheur dans l'un des sept formats suivants :

- Format E/S
- Format Entrée/Sortie des esclaves AS-Interface
- Format bloc fonction
- Format simple
- Format E/S réseau
- Format fonction pas à pas
- Format registre bits à décalage

**Format E/S**

Les objets entrée/sortie (%I, %Q, %IW et %QW) présentent un repère en trois parties (ex. : %IX.Y.Z) et apparaissent sous la forme suivante :

- type d'objet et repère de l'automate dans la partie supérieure gauche ;
- repère de l'expansion dans la partie supérieure centre ;
- voie d'E/S dans la partie supérieure droite.

Dans le cas d'une entrée (%I) et d'une sortie (%Q) simples, la lettre "U" pour un bit non forcé (unforced) ou la lettre "F" pour un bit forcé (forced) apparaît dans la partie inférieure gauche de l'écran. La valeur de forçage apparaît dans la partie inférieure droite de l'écran.

L'objet sortie %Q0.3.11 apparaît dans la zone d'affichage sous la forme suivante :

Q	0	3	1	1
F				1

**Format E/S des esclaves AS-Interface**

Les objets E/S des esclaves AS-Interface (%IA, %QA, %IWA et %QWA) présentent un repère en quatre parties (ex. : %IAx.y.z) et apparaissent sous la forme suivante :

- type d'objet dans la partie supérieure gauche ;
- repère du maître AS-Interface sur le bus d'expansion dans la partie centrale supérieure gauche ;
- repère de l'esclave sur le bus AS-Interface dans la partie centrale supérieure droite ;
- voie d'E/S de l'esclave dans la partie supérieure droite.

Dans le cas d'une entrée (%IA) et d'une sortie (%QA) simples, la lettre "U" pour un bit non forcé (unforced) ou la lettre "F" pour un bit forcé (forced) apparaît dans la partie inférieure gauche de l'écran. La valeur de forçage apparaît dans la partie inférieure droite de l'écran.

L'objet sortie %QA1.3A.2 apparaît dans la zone d'affichage sous la forme suivante :

QA	1	3A	2
F			1

**Format bloc  
fonction**

Les blocs fonction (%TM, %C, %FC, %VFC, %PLS, %PWM, %DR, %R et %MSGj) présentent un repère en deux parties comprenant le numéro de l'objet et le nom d'une variable ou d'un attribut. Ils apparaissent sous la forme suivante :

- nom du bloc fonction dans la partie supérieure gauche ;
- numéro (ou instance) du bloc fonction dans la partie supérieure droite ;
- variable ou attribut dans la partie inférieure gauche ;
- valeur de l'attribut dans la partie inférieure droite.

Dans l'exemple suivant, la valeur courante pour le temporisateur n° 123 est réglée sur 1 234.

T	M		1	2	3
V			1	2	3 4

---

**Format simple**

Un format simple est utilisé pour les objets %M, %MW, %KW, %MD, %KD, %MF, %KF, %S, %SW et %X :

- numéro de l'objet dans la partie supérieure droite ;
- valeur avec signe pour les objets dans la partie inférieure.

Dans l'exemple suivant, le mot mémoire n° 67 contient la valeur +123.

M	W		6	7
	+		1	2 3

---

**Format E/S  
réseau**

Les objets E/S réseau (%INW et %QNW) apparaissent dans la zone d'affichage sous la forme suivante :

- nom de l'objet dans la partie supérieure gauche ;
- repère de l'automate dans la partie supérieure centrale ;
- numéro de l'objet dans la partie supérieure droite ;
- valeur avec signe de l'objet dans la partie inférieure.

Dans l'exemple suivant, le premier mot d'entrée réseau de l'automate distant configuré à l'adresse distante n°2 a pour valeur -4.

I	N	W	2	0
	-			4



**Format fonction pas à pas**

Le format fonction pas à pas (%SC) affiche le numéro de l'objet et le bit de fonction pas à pas sous la forme suivante :

- nom et numéro de l'objet dans la partie supérieure gauche ;
- numéro du bit de fonction pas à pas dans la partie supérieure droite ;
- valeur du bit de fonction pas à pas dans la partie inférieure.

Dans l'exemple suivant, le bit n° 129 de la fonction pas à pas n° 3 est réglé sur 1.

S	C	3		1	2	9
						1

**Format registre bits à décalage**

Le registre bits à décalage (%SBR) apparaît dans la zone d'affichage sous la forme suivante :

- nom et numéro de l'objet dans la partie supérieure gauche ;
- numéro du bit de registre dans la partie supérieure droite ;
- valeur du bit de registre dans la partie inférieure droite.

Vous trouverez ci-après un exemple de l'affichage du registre bits à décalage n° 4.

S	B	R	4			9
						1

## Paramètres de port série




### Introduction

L'afficheur vous permet de visualiser les paramètres du protocole et de modifier les adresses de tous les ports série configurés à l'aide de TwidoSoft. Un maximum de deux ports série peut être utilisé. Dans l'exemple suivant, le premier port est configuré pour le protocole Modbus et porte l'adresse 123. Le second port est configuré en tant que liaison distante et porte l'adresse 4.

M	1	2	3
R			4

### Affichage et modification des paramètres d'un port série

Les automates Twido peuvent gérer un maximum de deux ports série. Pour visualiser les paramètres des ports série sur l'afficheur :

Etape	Action
1	Appuyez sur la touche  jusqu'à ce que l'écran Affichage des communications apparaisse. Une lettre, correspondant au paramètre de protocole du premier port (M, R ou A), sera affichée dans le coin supérieur gauche de l'afficheur.
2	Appuyez sur la touche MOD/ENTER pour passer en mode édition.
3	Appuyez sur la touche  jusqu'à ce que vous vous trouviez dans le champ à modifier.
4	Appuyez sur la touche  pour incrémenter la valeur de ce champ.
5	Répétez les étapes 3 et 4 jusqu'à ce que tous les paramètres de l'adresse aient été définis.
6	Appuyez sur la touche MOD/ENTER pour enregistrer les modifications apportées en mode édition ou sur ESC pour les ignorer.
7	

## Horloge calendrier

### Introduction

Les paramètres de date et d'heure ne peuvent être mis à jour depuis l'afficheur que si la cartouche optionnelle de l'horodateur (TWDXCPRTC) est installée sur votre automate Twido. Le mois apparaît dans la partie supérieure gauche de l'écran IHM. La valeur "RTC" figurera dans ce champ jusqu'à ce que des paramètres de date et d'heure valides aient été entrés. Le jour du mois apparaît dans la partie supérieure droite de l'écran. Cette heure est affichée au format dit "militaire". Les heures et les minutes sont affichées dans le coin inférieur droit de l'écran et sont séparées par la lettre "h". L'exemple suivant illustre ce qu'indiquerait l'écran, le 28 mars à 14:22.




M	A	R	2	8
			1	4 h 2 2

#### Note :

1. Les automates compacts TWDLCA•40DRF disposent d'un horodateur intégré.
2. Pour tous les autres automates, l'horloge calendrier et la correction RTC ne sont disponibles que lorsque la cartouche horodateur en option (TWDXCPRTC) est installée.

### Affichage et modification de l'horloge calendrier

Pour afficher et modifier l'horloge calendrier, procédez comme suit :

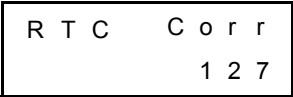
Etape	Action
1	Appuyez sur la touche  jusqu'à ce que l'écran de date et heure apparaisse. Le code du mois ("JAN" ou "FEV", par exemple) apparaît dans le coin supérieur gauche de la zone d'affichage. La mention "RTC" est affichée dans ce même coin tant que le mois n'a pas été défini.
2	Appuyez sur la touche MOD/ENTER pour passer en mode édition.
3	Appuyez sur la touche  jusqu'à ce que vous vous trouviez dans le champ à modifier.
4	Appuyez sur la touche  pour incrémenter la valeur de ce champ.
5	Répétez les étapes 3 et 4 jusqu'à ce que tous les paramètres de date et d'heure aient été définis.
6	Appuyez sur la touche MOD/ENTER pour enregistrer les modifications apportées en mode édition ou sur ECHAP pour les ignorer.

## Facteur de correction de l'horodateur

### Introduction




L'afficheur vous permet de visualiser et de modifier le facteur de correction de l'horodateur (RTC). Pour chaque module option horodateur (RTC), une valeur de correction permet de corriger les imprécisions du cristal du module horodateur. Ce facteur prend la forme d'un nombre entier sans signe, composé de trois chiffres, compris entre 0 et 127. Cette valeur apparaît dans le coin inférieur droit de l'afficheur.

L'exemple suivant illustre un facteur de correction de 127.



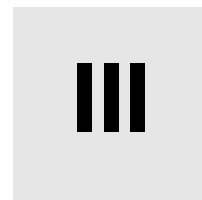
### Affichage et modification de la correction de l'horodateur

Pour afficher et modifier le facteur de correction de l'horodateur :

Etape	Action
1	Appuyez sur la touche  jusqu'à ce que l'écran Affichage du facteur de correction RTC apparaisse. "RTC Corr" s'affiche dans la ligne supérieure de l'afficheur.
2	Appuyez sur la touche MOD/ENTER pour passer en mode édition.
3	Appuyez sur la touche  jusqu'à ce que vous vous trouviez dans le champ à modifier.
4	Appuyez sur la touche  pour incrémenter la valeur de ce champ.
5	Répétez les étapes 3 et 4 jusqu'à ce que la valeur de correction du RTC ait été définie.
6	Appuyez sur la touche MOD/ENTER pour enregistrer les modifications apportées en mode édition ou sur ESC pour les ignorer.

---

# Description des langages Twido



---

## Présentation

### Objet de cette partie

Cette rubrique fournit des instructions d'utilisation des langages de programmation Grafcet, schéma à contacts et liste d'instructions permettant de créer des programmes pour des automates programmables Twido.

### Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
11	Langage schéma à contacts	263
12	Langage liste d'instructions	287
13	Grafcet	299



---

# Langage schéma à contacts

11

---

## Présentation

### Objet de ce chapitre

Ce chapitre décrit la programmation à l'aide du langage schéma à contacts.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Introduction aux schémas à contacts	264
Principes de programmation en langage schéma à contacts	266
Blocs de schémas à contacts	268
Éléments graphiques du langage schéma à contacts	271
Instructions spéciales OPEN et SHORT du langage schéma à contacts	274
Conseils de programmation	275
Réversibilité schéma à contacts/liste	280
Recommandations pour la réversibilité entre le langage schéma à contacts et le langage liste d'instructions	282
Documentation du programme	284

## Introduction aux schémas à contacts

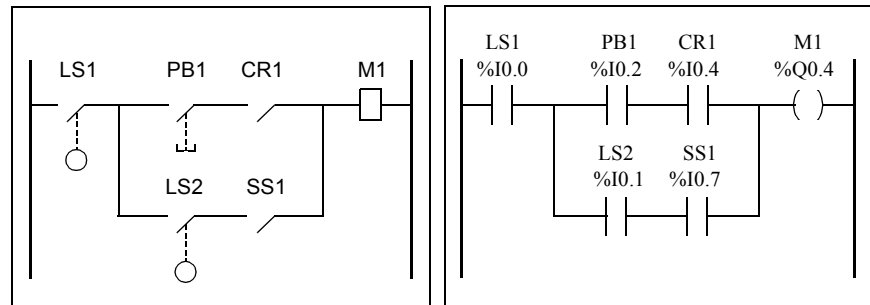
### Introduction

Les schémas à contacts utilisent la même représentation graphique que celle des circuits de relais en logique programmée, à ceci près que, dans un schéma à contacts :

- Toutes les entrées sont représentées par des symboles de contacts ( $\neg$  et  $\neg$ ).
- Toutes les sorties sont représentées par des symboles de bobines ( $\neg$  et  $\neg$ ).
- Les opérations numériques sont comprises dans le jeu d'instructions graphiques du schéma à contacts.

### Représentations de schémas à contacts correspondant aux circuits de relais

L'illustration suivante présente un schéma simplifié de câblage de relais en logique programmée, et son équivalent en langage schéma à contacts.



Circuit de relais en logique programmée

Schéma à contacts

Dans l'illustration précédente, toutes les entrées associées à un périphérique de commutation dans le circuit de relais en logique programmée sont représentées sous la forme de contacts dans le schéma à contacts. La bobine de sortie M1 du circuit logique de relais est représentée par un symbole de bobine dans le schéma à contacts. Les numéros des repères apparaissant au-dessus du symbole de chaque contact et de chaque bobine dans le schéma à contacts sont des références aux emplacements des connexions externes en entrée et en sortie vers l'automate.



## Réseaux schéma à contacts

Un programme en langage schéma à contacts est composé de "réseaux", représentant des ensembles d'instructions graphiques et apparaissant entre deux barres verticales. Les réseaux sont exécutés de manière séquentielle par l'automate.

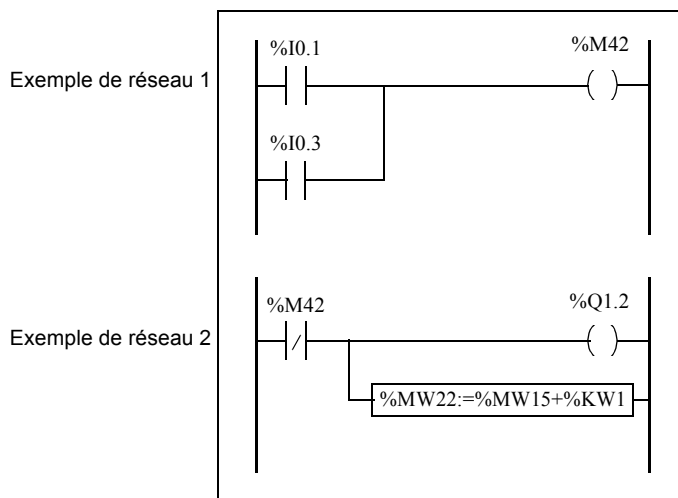
L'ensemble des instructions graphiques représente les fonctions suivantes :

- Entrées/sorties de l'automate (boutons de commande, capteurs, relais, voyants, etc.)
- Fonctions de l'automate (temporisateurs, compteurs, ...)
- Opérations mathématiques et logiques (addition, division, AND, XOR, etc.)
- Opérateurs de comparaison et autres opérations numériques ( $A < B$ ,  $A = B$ , décalage, rotation, etc.)
- Variables internes de l'automate (bits, mots, etc.)

Ces instructions sont disposées graphiquement selon des connexions verticales et horizontales, débouchant éventuellement sur une ou plusieurs sorties et/ou actions. Un réseau ne peut pas contenir plus d'un groupe d'instructions liées.

## Exemple de réseaux schéma à contacts

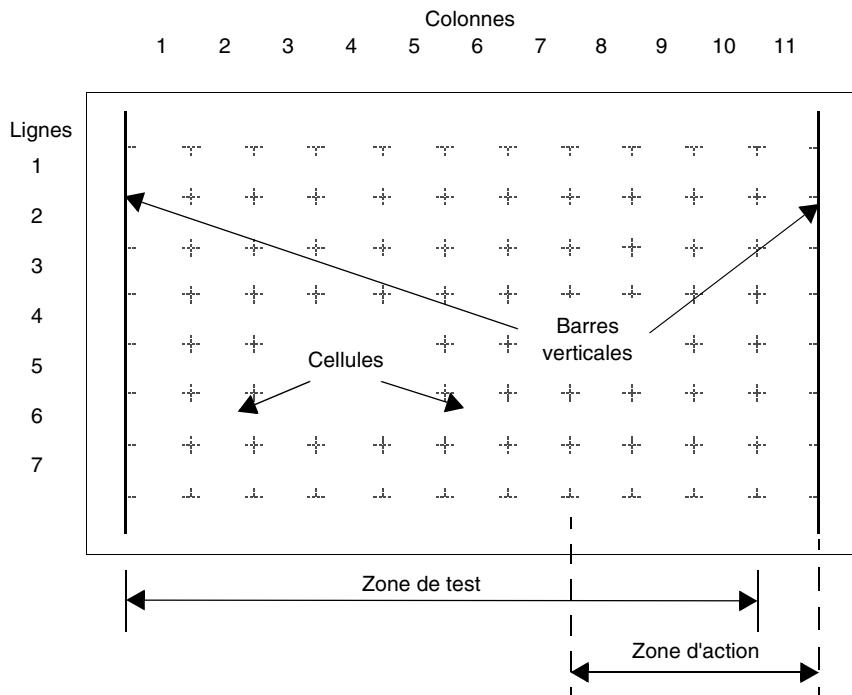
L'exemple suivant illustre un programme en langage schéma à contacts composé de deux réseaux.



## Principes de programmation en langage schéma à contacts

### Grille de programmation

Chaque réseau schéma à contacts se compose d'une grille comportant sept lignes et onze colonnes organisées en deux zones, comme l'indique l'illustration suivante :



### Zones de la grille

La grille de programmation en langage schéma à contacts est divisée en deux zones :

- Zone de test  
Contient les conditions testées avant d'effectuer des actions. Comprend les colonnes 1 à 7 et contient les contacts, les blocs fonctions et les blocs comparaisons.
- Zone d'action  
Contient la sortie ou l'opération qui sera effectuée en fonction des résultats des tests réalisés sur les conditions dans la zone de test. Comprend les colonnes 8 à 11 et contient les bobines et les blocs opérations.

**Saisie  
d'instructions  
dans la grille**

La grille de sept lignes sur onze colonnes que constitue le réseau schéma à contacts se lit à partir de la cellule située en haut à gauche. La programmation consiste à entrer des instructions dans les cellules de la grille. Les instructions de test, de comparaison et de fonctions sont entrées dans les cellules de la zone de test et sont justifiées à gauche. La logique du test permet d'assurer la continuité dans la zone d'action, où les bobines, les opérations numériques et les instructions de gestion du programme sont entrées et justifiées à droite.  
Le réseau est traité ou exécuté (tests effectués et sorties affectées) dans la grille de haut en bas et de gauche à droite.

---

**En-tête réseau**

Un en-tête apparaît directement au-dessus du réseau. Vous pouvez l'utiliser pour donner des informations sur la finalité logique du réseau. L'en-tête de réseau peut contenir les informations suivantes :

- le numéro du réseau ;
- des étiquettes (%Li) ;
- des déclarations de sous-programme (SRi:) ;
- le titre du réseau ;
- des commentaires sur le réseau.

Pour obtenir davantage d'informations sur l'utilisation d'un en-tête réseau pour documenter vos programmes, reportez-vous à la rubrique *Documentation du programme*, p. 284.

---

## Blocs de schémas à contacts

---

### Introduction

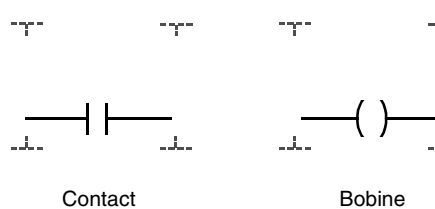
Les schémas à contacts se composent de blocs correspondant à des actions et/ou des fonctions d'un programme, telles que :

- des contacts
- des bobines
- des instructions de déroulement du programme
- des blocs fonctions
- des blocs comparaisons
- des blocs opérations

### Contacts, bobines et déroulement du programme

Les contacts, bobines et les instructions de déroulement du programme (sauts et appels) n'occupent qu'une seule cellule dans la grille de programmation du schéma à contacts. Les blocs fonctions, les blocs comparaisons et les blocs opérations peuvent en revanche occuper plusieurs cellules.

Les exemples suivants illustrent un contact et une bobine.

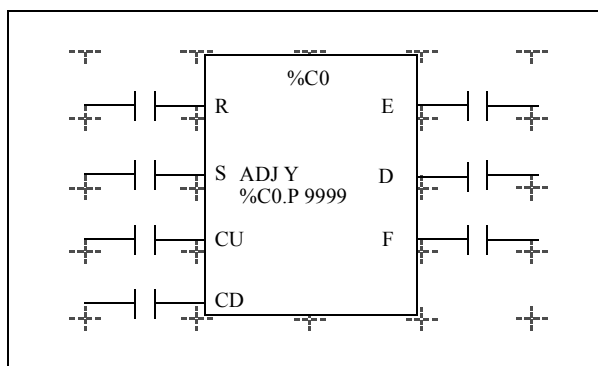


**Blocs fonctions**

Les blocs fonctions sont placés dans la zone de test de la grille de programmation. Le bloc doit figurer sur la première ligne ; aucune instruction de schéma à contacts ou aucune ligne de continuité ne peut apparaître au-dessus ou en dessous du bloc fonction. Les instructions de test du schéma à contacts mènent à l'entrée du bloc fonction, alors que les instructions de test et/ou les instructions d'action proviennent de la sortie du bloc.

Les blocs fonctions sont orientés de manière verticale et occupent deux colonnes sur quatre lignes dans la grille de programmation.

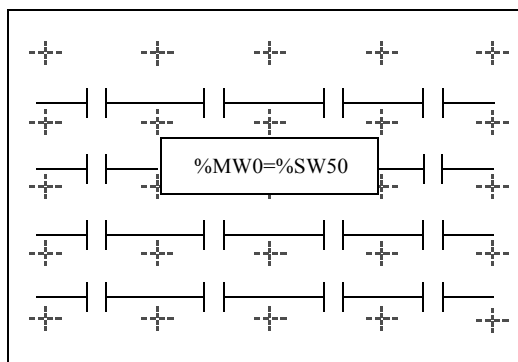
L'exemple suivant illustre un bloc fonction temporisateur.

**Blocs comparaisons**

Les blocs comparaisons sont placés dans la zone de test de la grille de programmation. Le bloc peut apparaître sur n'importe quelle ligne ou colonne de la zone de test. L'intégralité de l'instruction doit résider dans cette zone.

Les blocs comparaisons sont orientés de manière horizontale et occupent deux colonnes sur une ligne dans la grille de programmation.

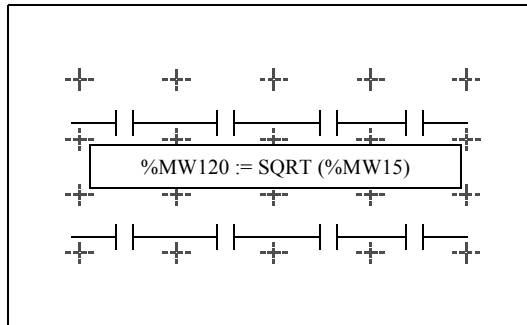
L'exemple suivant présente un bloc comparaison.



**Blocs opérations** Les blocs opérations sont placés dans la zone d'action de la grille de programmation. Le bloc peut apparaître sur n'importe quelle ligne de la zone d'action. L'instruction est justifiée à droite ; elle apparaît à droite et se termine dans la dernière colonne.

Les blocs opérations sont orientés de manière horizontale et occupent quatre colonnes sur une ligne dans la grille de programmation.

L'exemple suivant illustre un bloc opération.







## Éléments graphiques du langage schéma à contacts

### Introduction

Les instructions des schémas à contacts sont constituées d'éléments graphiques.



### Contacts

Les éléments graphiques des contacts sont programmés dans la zone de test et occupent une cellule (une ligne sur une colonne).

Nom	Élément graphique	Instruction	Fonction
Contact à ouverture		LD	Contact passant lorsque l'objet bit de contrôle se trouve à l'état 1.
Contact à fermeture		LDN	Contact passant lorsque l'objet bit de contrôle se trouve à l'état 0.
Contact de détection d'un front montant		LDR	Front montant : détecte le passage de 0 à 1 de l'objet bit de contrôle.
Contact de détection d'un front descendant		LDF	Front descendant : détecte le passage de 1 à 0 de l'objet bit de contrôle.






### Éléments de liaison

Les éléments de liaison graphique s'utilisent pour connecter les éléments graphiques de test et d'action.

Nom	Élément graphique	Fonction
Connexion horizontale		Relie en série les éléments graphiques de test et d'action entre les deux barres verticales.
Connexion verticale		Relie les éléments graphiques de test et d'action en parallèle.

**Bobines**


Les éléments graphiques des bobines sont programmés dans la zone d'action et occupent une cellule (une ligne sur une colonne).

Nom	Élément graphique	Instruction	Fonction
Bobine directe		ST	L'objet bit associé prend la valeur du résultat de la zone de test.
Bobine inverse		STN	L'objet bit associé prend la valeur du résultat inverse de la zone de test.
Bobine d'enclenchement		S	L'objet bit associé est réglé sur 1 lorsque le résultat de la zone de test est 1.
Bobine de déclenchement		R	L'objet bit associé est réglé sur 0 lorsque le résultat de la zone de test est 1.
Appel de saut ou de sous-programme	->>%Li ->>%SRi	JMP SR	Se connecte à une instruction portant une étiquette, en amont ou en aval.
Bobine dièse			Langage Grafcet. Utilisée lorsque la programmation des conditions de transition associées aux transitions provoque une permutation sur l'étape suivante.
Retour d'un sous-programme	<RET>	RET	Placé à la fin des sous-programmes pour retourner au programme principal.
Arrêt du programme	<END>	END	Définit la fin du programme.




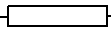
### Blocs fonction

Les éléments graphiques des blocs fonction sont programmés dans la zone de test et occupent quatre lignes sur deux colonnes (excepté les compteurs rapides (VFC), qui requièrent cinq lignes sur deux colonnes).

Nom	Élément graphique	Fonction
Temporisateurs, compteurs, registres, etc.		Chaque bloc fonction utilise les entrées et les sorties permettant la liaison aux autres éléments graphiques. Remarque : Les sorties des blocs fonction ne peuvent pas être connectées les unes aux autres (liaisons verticales).

### Blocs opérations et comparaisons

Les blocs comparaisons sont programmés dans la zone de test et les blocs opérations, dans la zone d'action.

Nom	Élément graphique	Fonction
Bloc comparaison		Compare deux opérands. La sortie prend la valeur 1 lorsque le résultat est vérifié. Taille : Une ligne sur deux colonnes
Bloc opération		Effectue des opérations arithmétiques et logiques. Taille : Une ligne sur quatre colonnes

## Instructions spéciales OPEN et SHORT du langage schéma à contacts

### Introduction

Les instructions OPEN et SHORT permettent de déboguer rapidement et simplement des programmes en langage schéma à contacts. Ces instructions spéciales modifient la logique d'un réseau, soit en raccourcissant, soit en ouvrant la continuité d'un réseau, conformément aux explications fournies dans le tableau suivant.

Instruction	Description	Instruction en langage liste d'instructions
OPEN	Crée un arrêt dans la continuité d'un réseau schéma à contacts, et ce, quels que soient les résultats de la dernière opération logique.	AND 0
SHORT	Permet à la continuité de traverser le réseau schéma à contacts, et ce, quels que soient les résultats de la dernière opération logique.	OR 1

En langage liste d'instructions, les instructions OR et AND sont utilisées pour créer les instructions OPEN et SHORT à l'aide des valeurs immédiates respectives de 0 et 1.

### Exemples

Les exemples suivants illustrent l'utilisation des instructions SHORT et OPEN.

```
graph LR
    subgraph Network1 [ ]
        direction LR
        I01["%I0.1"] --- Q15["%Q1.5"]
        Q15 --- I09["%I0.9"]
        I09 --- M3["%M3"]
        M3 --- OPEN["OPEN"]
        OPEN --- Q01["%Q0.1"]
    end
    subgraph Network2 [ ]
        direction LR
        I09["%I0.9"] --- SHORT["SHORT"]
        SHORT --- Q16["%Q1.6"]
    end
```

```
LD    %I0.1
OR    %Q1.5
ANDN  %M3
AND   0
ST    %Q0.1
LD    %I0.9
OR    1
ST    %Q1.6
```

---

## Conseils de programmation

---

**Gestion des sauts de programme**

Utilisez les sauts de programme avec la plus grande précaution, car ils peuvent être à l'origine de boucles qui ralentiront considérablement les opérations de scrutation. Evitez d'insérer des sauts pointant vers des instructions situées en amont. (Une instruction en amont apparaît avant un saut dans un programme. A l'inverse, une instruction en aval apparaît après un saut dans un programme).

---

**Programmation des sorties**

Les bits de sortie, tout comme les bits internes, ne doivent être modifiés qu'une seule fois dans le programme. Pour les bits de sortie, seule la dernière valeur scrutée est prise en compte lors de la mise à jour des sorties.

---

**Utilisation de capteurs d'arrêt d'urgence à liaison directe**

Les capteurs utilisés en cas d'arrêt d'urgence ne doivent pas être gérés par l'automate. Ces capteurs doivent être raccordés directement aux sorties correspondantes.

---

**Gestion des reprises de l'alimentation**

Conditionner une reprise secteur à une opération manuelle. Un redémarrage automatique peut entraîner un fonctionnement non désiré de l'installation (utilisez les bits système %S0, %S1 et %S9).

---

**Gestion de l'heure et des blocs horodateur**

Il est nécessaire de vérifier l'état du bit système %S51, qui signale d'éventuels défaut de l'horodateur.

---

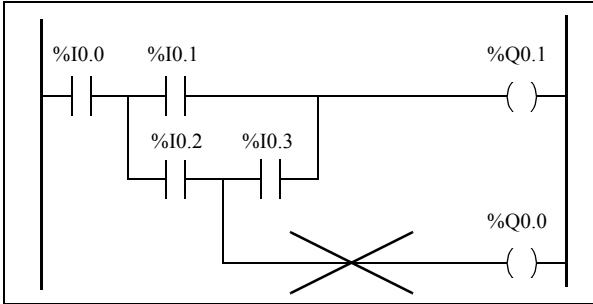
**Vérification de la syntaxe et recherche d'erreurs**

Lors de la saisie d'un programme, TwidoSoft vérifie la syntaxe de ses instructions et opérandes, ainsi que leur association.

---

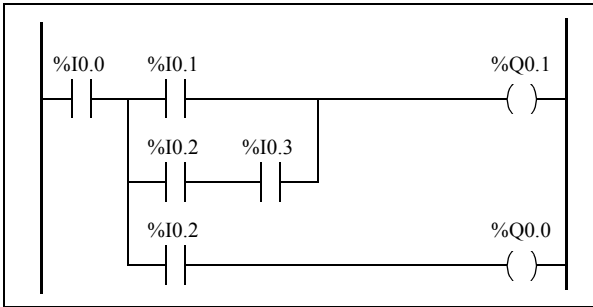
Remarques complémen-  
taires sur  
l'utilisation des  
parenthèses

Les opérations d'affectation ne doivent pas être placés entre parenthèses :



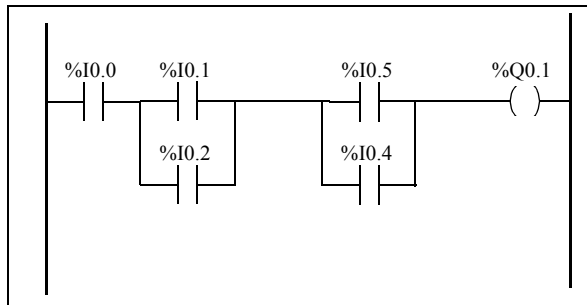
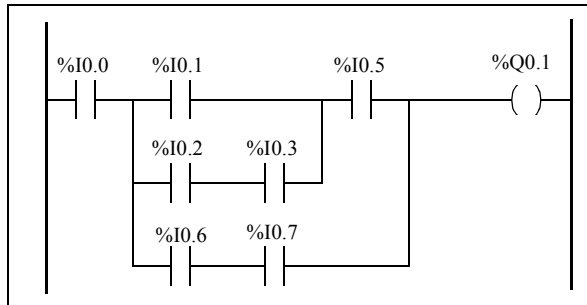
```
LD    %I0.0
AND   %I0.1
      ( )
OR(   %I0.2
ST  %Q0.0
AND %I0.3
)
ST    %Q0.1
```

Afin d'effectuer la fonction correspondante, les équations suivantes doivent être programmées :

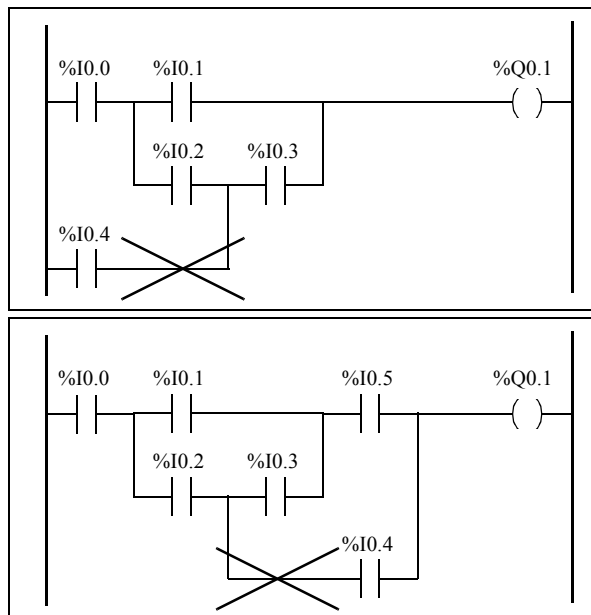


```
LD    %I0.0
MPS
AND(  %I0.1
OR(   %I0.2
AND   %I0.3
)
)
ST    %Q0.1
MPP
AND   %I0.2
ST    %Q0.0
```

Si plusieurs mises en parallèle de contact sont effectuées, elles devront être imbriquées les unes dans les autres ou complètement dissociées :



Les schémas suivants ne peuvent pas être programmés :



Afin d'exécuter les schémas équivalents, modifiez-les comme illustré ci-dessous :

	<pre>LD      %I0.0 AND(    %I0.1 OR(      %I0.2 AND      %I0.3 ) ) OR(      %I0.4 AND      %I0.3 ) ST      %Q0.1</pre>
	<pre>LD      %I0.0 AND(    %I0.1 OR(      %I0.2 AND      %I0.3 ) AND      %I0.5 OR(      %I0.2 AND      %I0.4 ) ) ST      %Q0.1</pre>

## Réversibilité schéma à contacts/liste

---

### Introduction

La fonctionnalité de réversibilité du logiciel de programmation TwidoSoft permet de convertir des programmes par schémas à contacts en programmes par listes d'instructions, et vice versa.

Les préférences utilisateur réglées dans TwidoSoft permettent de choisir la méthode d'affichage par défaut des programmes : soit au format liste, soit au format schéma à contacts. TwidoSoft permet également de basculer entre les affichages par liste et par schéma à contacts.

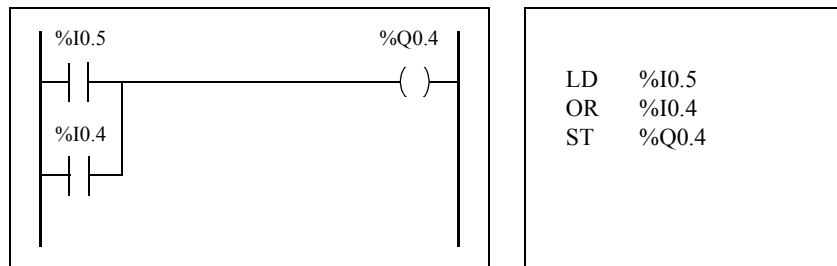
---

### Qu'est-ce que la "réversibilité" ?

Pour bien comprendre à quoi correspond la fonction de réversibilité du programme, il convient d'examiner avec attention les relations existant entre le réseau d'un schéma à contacts et la séquence de la liste d'instructions correspondante :

- **Réseau de schéma à contacts** : ensemble d'instructions par schémas à contacts formant une expression logique.
- **Séquence de liste** : ensemble d'instructions d'un programme par listes, correspondant aux instructions par schémas à contacts et relatif à la même expression logique.

L'illustration suivante présente un réseau de schéma à contacts courant, ainsi que la logique du programme équivalente, exprimée sous la forme d'une liste d'instructions.



Un programme d'application est stocké en interne sous la forme d'une liste d'instructions, et ce, que le programme ait été rédigé en langage par schémas à contacts ou par listes. TwidoSoft utilise les similarités de structure de programme existant entre les deux langages, ainsi que l'image liste interne du programme pour l'afficher soit sous la forme d'une liste d'instructions (forme élémentaire), soit de manière graphique, sous la forme d'un schéma à contacts, en fonction des préférences sélectionnées par l'utilisateur.

---



**Garantie de réversibilité**

Tout programme créé sous forme de schéma à contacts peut être converti en une liste d'instructions. En revanche, certaines logiques du langage par listes ne peuvent pas être converties en langage par schémas à contacts. Pour garantir une réversibilité totale entre le langage par listes et le langage par schémas à contacts, il est important d'observer les directives présentées à la section *Recommandations pour la réversibilité entre le langage schéma à contacts et le langage liste d'instructions*, p. 282.

---

## Recommandations pour la réversibilité entre le langage schéma à contacts et le langage liste d'instructions

---

### Instructions requises pour la réversibilité

La structure d'un bloc fonction réversible dans le langage liste d'instructions requiert l'utilisation des instructions suivantes :

- **BLK** marque le début du bloc et définit le début du réseau, ainsi que celui de la portion d'entrée dans le bloc.
- **OUT\_BLK** marque le début de la portion de sortie du bloc.
- **END\_BLK** marque la fin du bloc et du réseau.

Il n'est pas nécessaire d'utiliser des instructions de blocs fonctions réversibles pour un programme liste d'instructions qui fonctionne correctement. Certaines instructions permettent une programmation liste d'instructions non réversible. Pour obtenir des informations complètes sur la programmation liste d'instructions non réversible de blocs fonctions, reportez-vous à la rubrique *Principes de programmation de blocs fonction standards*, p. 335.

---

### Instructions sans équivalences à éviter

Évitez d'utiliser certaines instructions en langage liste ou certaines associations d'instructions et d'opérandes, pour lesquelles les schémas à contacts ne possèdent pas d'équivalents. Par exemple, l'instruction N (permettant d'inverser la valeur de l'accumulateur booléen) n'a pas d'équivalent dans le langage schémas à contacts. Le tableau suivant répertorie toutes les instructions de programmation liste d'instructions qui ne s'inversent pas dans le langage schéma à contacts :

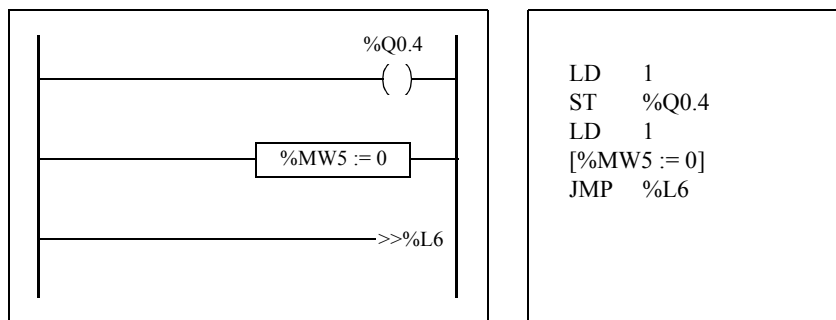
Instruction par liste	Opérande	Description
JMPCN	%Li	Not saut conditionnel
N	aucun	Négation (Not)
ENDCN	aucun	Not fin conditionnelle

---

## Réseaux inconditionnels

La programmation des réseaux inconditionnels requiert également l'application des recommandations de programmation liste d'instructions suivantes pour que la réversibilité liste d'instructions/schéma à contacts puisse s'opérer. Les réseaux inconditionnels ne sont soumis à aucun test ou à condition. Les sorties ou les instructions d'action sont toujours activées ou exécutées.

Le diagramme suivant présente des exemples de réseaux inconditionnels, ainsi que la séquence en langage liste d'instructions équivalente.



Vous noterez que chacune des séquences liste d'instructions inconditionnelles ci-dessus commence par une instruction de chargement suivie d'un 1, excepté pour l'instruction JMP. Cette combinaison écrit la valeur de l'accumulateur booléen à 1, et met par conséquent la bobine (instruction de stockage) à 1 et %MW5 à 0 lors de chaque scrutation du programme. L'exception est l'instruction de saut liste inconditionnel (JMP %L6), qui est exécutée quelle que soit la valeur de l'accumulateur et ne nécessite pas l'écriture de l'accumulateur à un.

## Réseau schéma à contacts / liste d'instructions

Si un programme liste d'instructions qui n'est pas totalement réversible est inversé, les parties réversibles sont affichées dans la visualisation par schémas à contacts et celles qui sont irréversibles sont affichées sur les réseaux schéma à contacts en liste d'instructions.

Un réseau schéma à contacts en liste d'instructions fonctionne exactement comme un petit éditeur liste d'instructions. Il permet en effet à l'utilisateur de visualiser et de modifier les parties irréversibles d'un programme schéma à contacts.

## Documentation du programme

---

### Documentation de votre programme

Vous pouvez documenter votre programme en y ajoutant des commentaires à l'aide des éditeurs liste d'instructions et schéma à contacts :

- Dans l'éditeur de listes, des commentaires de lignes vous permettent de documenter votre programme. Ces commentaires peuvent figurer sur la même ligne que les instructions de programmation, ou sur des lignes individuelles distinctes.
- Dans l'éditeur de schémas à contacts, des en-têtes réseau vous permettent de documenter votre programme. Ces en-têtes se situent juste au-dessus du réseau.

Le logiciel de programmation TwidoSoft utilise ces commentaires à des fins de réversibilité. Lors de la conversion d'un programme par listes en programme par schémas à contacts, TwidoSoft utilise certains des commentaires liste pour créer un en-tête réseau. Pour ce faire, les commentaires insérés entre les séquences de liste sont utilisés comme en-têtes réseau.

---

### Exemple de commentaires de ligne de liste

L'exemple suivant illustre un programme par listes comportant des commentaires de lignes.

```
---- ( * TITRE DE L'EN-TETE DU RESEAU 0 * )
---- ( * PREMIER COMMENTAIRE DE L'EN-TETE DU RESEAU 0 * )
---- ( * DEUXIEME COMMENTAIRE DE L'EN-TETE DU RESEAU 0 * )
  0 LD %I0.0 ( * COMMENTAIRE DE LIGNE * )
  1 OR %I0.1 ( * LIGNE DE COMMENTAIRE IGNOREE LORS DE LA CONVERSION EN
SCHEMA À CONTACTS * )
  2 ANDM %M10
  3 ST M101
---- ( * EN-TETE DU RESEAU 1 * )
---- ( * CE RESEAU CONTIENT UNE ETIQUETTE * )
---- ( * DEUXIEME COMMENTAIRE DE L'EN-TETE DU RESEAU 1 * )
---- ( * TROISIEME COMMENTAIRE DE L'EN-TETE DU RESEAU 1 * )
---- ( * QUATRIEME COMMENTAIRE DE L'EN-TETE DU RESEAU 1 * )
  4 %L5:
  5 LD %M101
  6 [ %MW20 := %KW2 * 16]
---- ( * CE RESEAU NE CONTIENT QUE LE TITRE D'UN EN-TETE * )
  7 LD %Q0.5
  8 OR %I0.3
  9 ORR I0.13
 10 ST %Q0.5
```

---

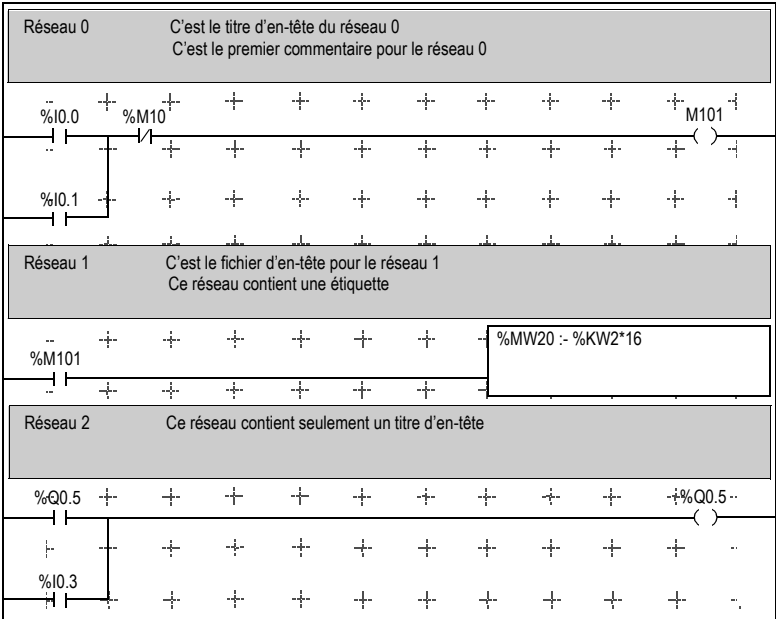
**Conversion de commentaires de liste en en-tête réseau de schéma à contacts**

Lorsque qu'un programme par listes est converti en programme par schémas à contacts, les commentaires de ligne de liste sont affichés dans l'éditeur de schémas à contacts selon les règles suivantes :

- Le premier commentaire figurant sur une ligne individuelle est utilisé comme en-tête réseau.
- Les commentaires suivants sont utilisés pour former le corps du réseau.
- Lorsque les lignes du corps de l'en-tête sont toutes remplies, les commentaires de ligne compris entre les séquences de liste sont ignorés, tout comme les autres commentaires situés dans des lignes de liste et qui contiennent également des instructions.

**Exemple de commentaires d'en-têtes réseau**

L'exemple suivant illustre un programme par schémas à contacts comportant des commentaires d'en-têtes réseau.



**Conversion de commentaires de schémas à contacts en commentaires de listes**

Lorsqu'un schéma à contacts est converti en une liste d'instructions, les commentaires d'en-têtes réseau sont affichés dans l'éditeur de listes selon les règles suivantes :

- Tous les commentaires d'en-tête réseau sont insérés entre les séquences de liste associées.
  - Toutes les étiquettes (%Li:) ou les déclarations de sous-programme (SRi:) sont placées sur la ligne suivant l'en-tête et précédant immédiatement la séquence de liste.
  - Si le programme avait déjà été converti du format liste au format schéma à contacts, tous les commentaires précédemment ignorés seront de nouveau affichés dans l'éditeur de listes.
-

---

# Langage liste d'instructions

# 12

---

## Présentation

### Objet de ce chapitre

Ce chapitre décrit la programmation à l'aide du langage liste d'instructions.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Vue d'ensemble des programmes en langage liste d'instructions	288
Fonctionnement des listes d'instructions	290
Instructions en langage liste d'instructions	291
Utilisation de parenthèses	294
Instructions de pile (MPS, MRD, MPP)	297

## Vue d'ensemble des programmes en langage liste d'instructions

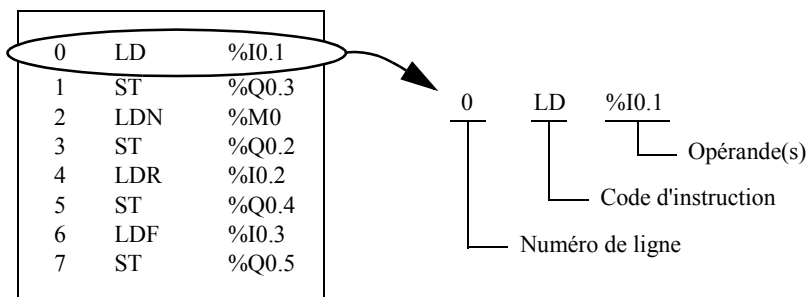
### Introduction

Un programme écrit en langage liste d'instructions est constitué d'une série d'instructions exécutées en séquence par l'automate. Chaque instruction est représentée par une seule ligne de code et se compose de trois éléments :

- Numéro de ligne
- Code d'instruction
- Opérande(s)

### Exemple de programme liste d'instructions

L'illustration suivante est un exemple de programme liste d'instructions.



### Numéro de ligne

Les numéros de ligne sont générés automatiquement lorsque vous saisissez une instruction. Les lignes vides et les lignes de commentaires n'ont pas de numéro de ligne.



**Code d'instruction**

Le code d'instruction est un symbole désignant un opérateur qui identifie l'opération à effectuer à l'aide des opérandes. Les opérateurs types spécifient les opérations booléennes et numériques.

Par exemple, dans l'échantillon de programme présenté ci-dessus, LD est l'abréviation de LOAD en code d'instruction. L'instruction LOAD place (charge) la valeur de l'opérande %I0.1 dans un registre interne nommé accumulateur.

Il existe deux types d'instructions de base :

- Instructions de test

Il s'agit de tests des conditions ou résultat d'équation nécessaires à l'accomplissement d'une action. Par exemple, LOAD (LD) et AND.

- Instructions d'action

Elles permettent d'effectuer les actions autorisées lorsque les conditions de test sont remplies. Par exemple, des instructions d'affectation telles que STORE (ST) et RESET (R).

---

**Opérande**

Un opérande est un nombre, un repère ou un symbole représentant une valeur qu'un programme peut manipuler au sein d'une instruction. Par exemple, dans l'échantillon de programme présenté ci-dessus, l'opérande %I0.1 est un repère auquel on a affecté la valeur d'une entrée de l'automate. Une instruction peut avoir entre zéro et trois opérandes selon le type de code d'instruction.

Les opérandes peuvent représenter les éléments suivants :

- les entrées/sorties de l'automate, telles que les capteurs, boutons poussoirs et relais ;
  - les fonctions système prédéfinies, telles que les temporisateurs et les compteurs ;
  - les opérations arithmétiques, logiques, de comparaisons et numériques ;
  - les variables internes de l'automate, telles que les bits et les mots.
-

## Fonctionnement des listes d'instructions

---

### Introduction

Les listes d'instructions ne possèdent qu'un seul opérande explicite, l'autre étant implicite. L'opérande implicite correspond à la valeur de l'accumulateur booléen. Par exemple, dans l'instruction LD %I0.1, %I0.1 est l'opérande explicite. Un opérande implicite est stocké dans l'accumulateur et se voit écrasé par la valeur de %I0.1.

### Fonctionnement

Une instruction en langage liste d'instructions exécute une opération spécifiée sur le contenu de l'accumulateur et sur l'opérande explicite, puis remplace le contenu de l'accumulateur par le résultat obtenu. Par exemple, l'opération AND %I1.2 effectue un AND logique entre le contenu de l'accumulateur et celui de l'entrée 1.2 et remplace le contenu de l'accumulateur par ce résultat.

L'ensemble des instructions booléennes, à l'exception des instructions de chargement, de stockage et les instructions NOT, fonctionnent avec deux opérandes. La valeur des deux opérandes peut être True ou False et l'exécution des instructions par le programme génère une valeur unique : soit True, soit False. Les instructions de chargement placent la valeur de l'opérande dans l'accumulateur, tandis que les instructions de stockage transfèrent la valeur de l'accumulateur vers l'opérande. L'instruction NOT ne comporte aucun opérande explicite et a seulement pour effet d'inverser l'état de l'accumulateur.

### Instructions en langage liste d'instructions prises en charge

Le tableau suivant représente quelques instructions en langage liste :

Type d'instruction	Exemple	Fonction
Instruction sur bit	LD %M10	Lit le bit interne %M10
Instruction sur bloc	IN %TM0	Démarre le temporisateur %TM0
Instruction sur mot	[%MW10 := %MW50+100]	Opération d'addition
Instruction sur programme	SR5	Appelle le sous-programme n°5
Instruction Grafcet	-* -8	Etape n°8

# Instructions en langage liste d'instructions

## Introduction

Le langage liste d'instructions comprend les types d'instructions suivants :

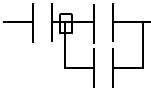
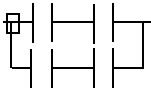
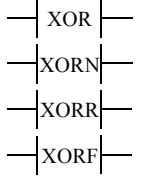
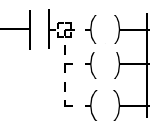
- Instructions sur test
- Instructions sur action
- Instructions sur bloc fonction

Ce sous-chapitre identifie et décrit les instructions Twido de programmation en langage liste d'instructions.

## Instructions sur test





Le tableau suivant décrit les instructions sur test du langage liste d'instructions.

Nom	Élément graphique correspondant	Fonction
LD		Le résultat booléen correspond à l'état de l'opérande.
LDN		Le résultat booléen correspond à l'état inversé de l'opérande.
LDR		Le résultat booléen prend la valeur 1 lorsque le passage de l'opérande (front montant) de 0 à 1 est détecté.
LDF		Le résultat booléen devient 1 lorsque le passage de l'opérande (front descendant) de 1 à 0 est détecté.
AND		Le résultat booléen est égal à la logique AND entre le résultat booléen de l'instruction précédente et l'état de l'opérande.
ANDN		Le résultat booléen est égal à la logique AND entre le résultat booléen de l'instruction précédente et l'état inversé de l'opérande.
ANDR		Le résultat booléen est égal à la logique AND entre le résultat booléen de l'instruction précédente et la détection du front montant de l'opérande (1 = front montant).
ANDF		Le résultat booléen est égal à la logique AND entre le résultat booléen de l'instruction précédente et la détection du front descendant de l'opérande (1 = front descendant).
OR		Le résultat booléen est égal à la logique OR entre le résultat booléen de l'instruction précédente et l'état de l'opérande.

Nom	Élément graphique correspondant	Fonction
AND(		Logique AND (8 niveaux de parenthèses)
OR(		Logique OR (8 niveaux de parenthèses)
XOR, XORN, XORR, XORF		OR exclusif
MPS MRD MPP		Commutation vers les bobines
N	-	Négation (NOT)

Instructions sur action


Le tableau suivant décrit les instructions sur action du langage liste d'instructions.

Nom	Élément graphique correspondant	Fonction
ST		L'opérande associé prend la valeur du résultat de la zone de test.
STN		L'opérande associé prend la valeur inversée du résultat de la zone de test.
S		L'opérande associé est réglé sur 1 lorsque le résultat de la zone de test est 1.
R		L'opérande associé est réglé sur 0 lorsque le résultat de la zone de test est 1.

Nom	Élément graphique correspondant	Fonction
JMP	->>%Li	Se connecte inconditionnellement à une séquence portant une étiquette, en amont ou en aval.
SRn	->>%SRi	Connexion au début d'un sous-programme.
RET	<RET>	Retour d'un sous-programme.
END	<END>	Fin de programme.
ENDC	<ENDC>	Fin du programme conditionné avec un résultat booléen de 1.
ENDCN	<ENDCN>	Fin du programme conditionné avec un résultat booléen de 0.

### Instructions de blocs fonction

Le tableau suivant décrit les instructions sur bloc fonction du langage liste d'instructions.

Nom	Élément graphique correspondant	Fonction
Temporisateurs, compteurs, registres, etc.		<p>Il existe des instructions de régulation de bloc pour chaque bloc fonction.</p> <p>Une forme structurée est utilisée pour raccorder directement les entrées et les sorties du bloc.</p> <p><b>Remarque :</b> Les sorties des blocs fonction ne peuvent pas être connectées les unes aux autres (liaisons verticales).</p>

## Utilisation de parenthèses

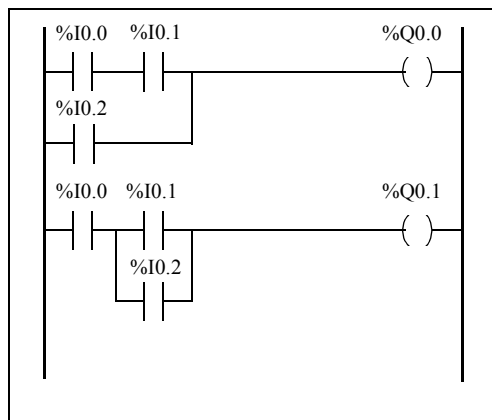
### Introduction

Dans les instructions logiques AND et OR, les parenthèses permettent de spécifier des divergences dans des schémas à contacts. Les parenthèses sont associées à des instructions, de la manière suivante :

- L'ouverture des parenthèses est associée à l'instruction AND ou OR.
- La fermeture des parenthèses correspond à une instruction requise pour chaque parenthèse ouverte.

### Exemple d'utilisation d'une instruction AND

Les schémas suivants illustrent l'utilisation des parenthèses dans une instruction AND : AND(...).

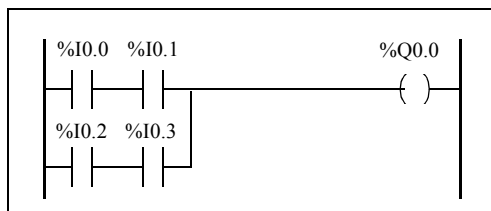


```
LD    %I0.0
AND   %I0.1
OR    %I0.2
ST    %Q0.0
```

```
LD    %I0.0
AND(  %I0.1
OR    %I0.2
)
ST    %Q0.1
```

### Exemple d'utilisation d'une instruction OR

Les schémas suivants illustrent l'utilisation des parenthèses dans une instruction OR : OR(...).



```
LD    %I0.0
AND   %I0.1
OR(   %I0.2
AND   %I0.3
)
ST    %Q0.0
```

**Modificateurs**

Le tableau suivant répertorie les modificateurs pouvant être associés à des parenthèses.

Modifica- teur	Fonction	Exemple
N	Négation	AND(N ou OR(N
F	Front descendant	AND(F ou OU(F
R	Front montant	AND(R ou OU(R
[	Comparaison	Reportez-vous à la rubrique <i>Instructions de comparaison</i> , p. 365

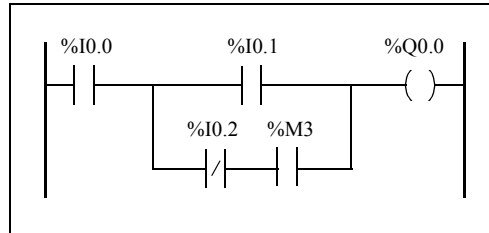
**Imbrication de parenthèses**

Il est possible d'imbriquer un maximum de huit niveaux de parenthèses. Veuillez appliquer les règles suivantes lors de l'imbrication de parenthèses :

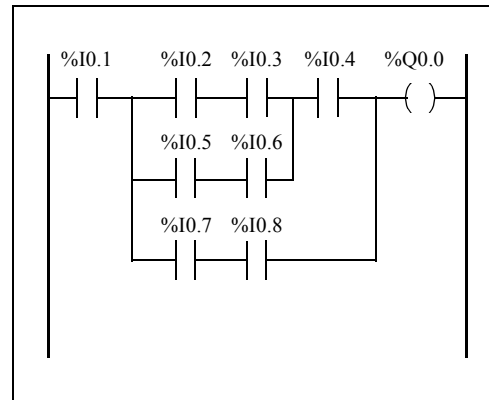
- Chaque parenthèse ouverte doit être obligatoirement refermée.
- Les étiquettes (%Li:), les sous-programmes (SRi:), les instructions de saut (JMP) et les instructions de bloc fonction ne doivent pas être placés dans des expressions comprises entre parenthèses.
- Les instructions de stockage ST, STN, S et R ne doivent pas être programmées entre parenthèses.
- Les instructions de pile MPS, MRD et MPP ne peuvent pas être utilisées entre parenthèses.

## Exemples d'imbrication de parenthèses

Les schémas suivants illustrent l'imbrication de parenthèses.



```
LD      %I0.0
AND(    %I0.1
OR(N    %I0.2
AND     %M3
)
)
ST      %Q0.0
```



```
LD      %I0.1
AND(    %I0.2
AND     %I0.3
OR(     %I0.5
AND     %I0.6
)
AND     %I0.4
OR(     %I0.7
AND     %I0.8
)
)
ST      %Q0.0
```



## Instructions de pile (MPS, MRD, MPP)

### Introduction

Les instructions de pile permettent de traiter le routage vers des bobines .Les instructions MPS, MRD et MPP utilisent une zone de stockage temporaire appelée « pile ». Cette pile peut stocker un maximum de huit expressions booléennes.

**Note :** Ces instructions ne peuvent pas être utilisées dans une expression comprise entre parenthèses.

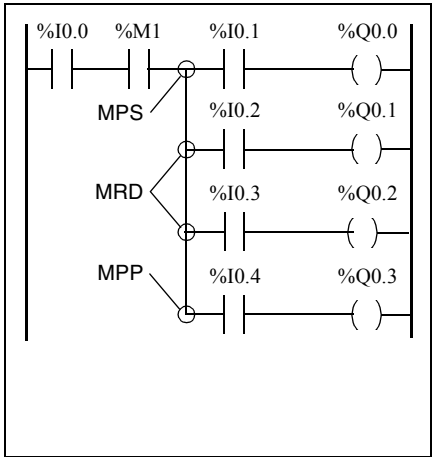
### Fonctionnement des instructions de pile

Le tableau suivant décrit le fonctionnement des trois instructions de pile.

Instruction	Description	Fonction
MPS	Abréviation de Memory Push onto Stack (Mise en mémoire sur la pile)	Stocke le résultat de la dernière instruction logique (contenu de l'accumulateur) en haut de la pile. Ceci a pour effet de décaler les autres valeurs de la pile vers le bas.
MRD	Abréviation de Memory Read from stack (Lecture mémoire depuis la pile)	Lit la valeur stockée en haut de la pile et la transmet à l'accumulateur.
MPP	Abréviation de Memory Pop from Stack (Extraction mémoire depuis la pile)	Lit la valeur située dans le haut de la pile, la transmet à l'accumulateur et déplace les autres valeurs de la pile vers le haut.

Exemples  
d'instructions de  
pile

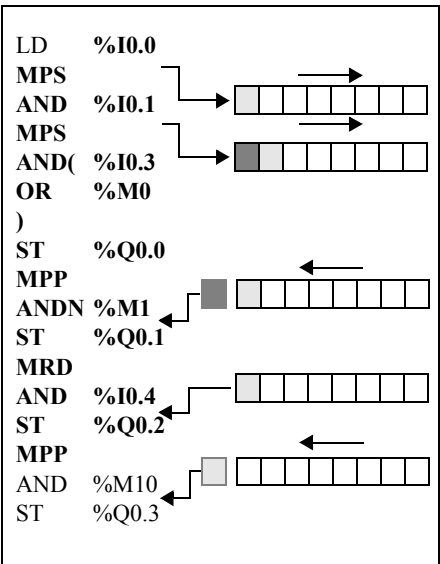
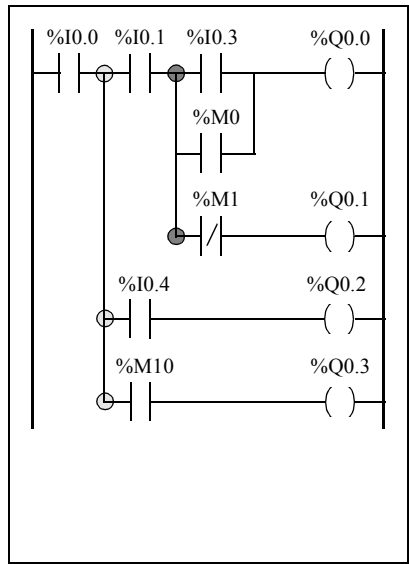
Les schémas suivants illustrent l'utilisation d'instructions de pile.



```
LD      %I0.0
AND     %M1
MPS
AND     %I0.1
ST      %Q0.0
MRD
AND     %I0.2
ST      %Q0.1
MRD
AND     %I0.3
ST      %Q0.2
MPP
AND     %I0.4
ST      %Q0.3
```

Exemples du  
fonctionnement  
de la pile

Les schémas suivants illustrent le fonctionnement des instructions de pile.



---

## Présentation

### Objet de ce chapitre

Ce chapitre décrit la programmation à l'aide du langage Grafcet.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description des instructions Grafcet	300
Description de la structure d'un programme Grafcet	305
Actions associées aux étapes Grafcet	308

## Description des instructions Grafcet

---

### Introduction

Les instructions Grafcet de TwidoSoft offrent une méthode simple de traduction de séquences de contrôle (graphe Grafcet).

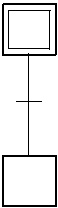



Le nombre maximum d'étapes Grafcet dépend du type d'automate Twido. Le nombre d'étapes pouvant être activées simultanément est uniquement limité par le nombre total d'étapes.

Pour les automates TWDLCAA10DRF et TWDLCAA16DRF, vous disposez des étapes 1 à 62. Les étapes 0 et 63 sont réservées pour le traitement antérieur et postérieur. Pour tous les autres automates, vous disposez des étapes 1 à 95.

---

## Instructions Grafcet

Le tableau suivant répertorie toutes les instructions et les objets requis pour la programmation d'un graphe Grafcet.

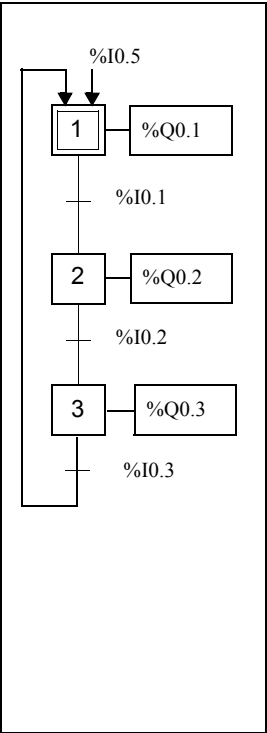
Représentation graphique (1)	Transcription en langage TwidoSoft	Fonction
Illustration : 		
<b>Étape initiale</b>	<code>=*= i</code>	Lance l'étape initiale (2).
<b>Transition</b>	<code># i</code>	Active l'étape i après avoir désactivé l'étape courante.
<b>Etape</b>	<code>-*- i</code>	Lance l'étape i et valide la transition associée (2).
	<code>#</code>	Désactive l'étape courante sans activer d'autre étape.
	<code>#Di</code>	Désactive l'étape i et l'étape courante.
	<code>=*= POST</code>	Lance le traitement postérieur et termine le traitement séquentiel.
	<code>%Xi</code>	Bit associé à l'étape i. Peut être testé et écrit (le nombre maximum d'étapes dépend de l'automate).
	<code>LD %Xi, LDN %Xi, AND %Xi, ANDN %Xi, OR %Xi, ORN %Xi, XOR %Xi, XORN %Xi</code>	Teste l'activité de l'étape i.
	<code>S %Xi</code>	Active l'étape i.
	<code>R %Xi</code>	Désactive l'étape i.

(1) La représentation graphique n'est pas prise en charge.

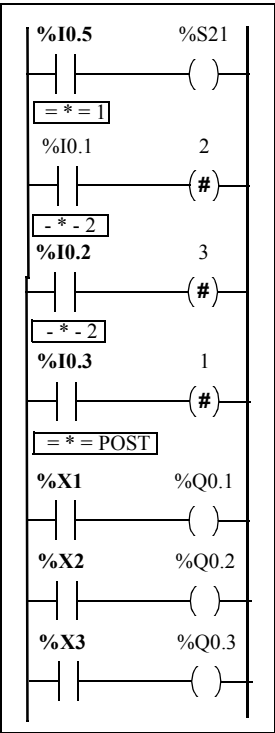
(2) La première étape `=*=i` ou `-*-i` écrite indique le lancement du traitement séquentiel et, par conséquent, la fin du pré-traitement.

Exemples  
Grafcet

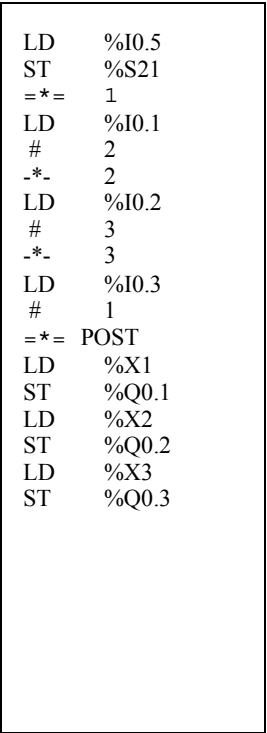
Séquence linéaire :



Non pris en charge

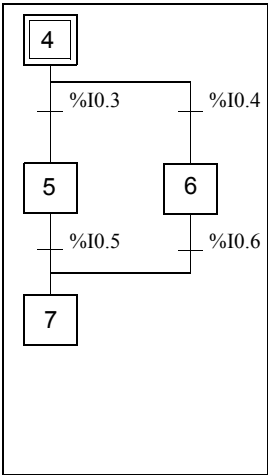


Programme schéma  
à contacts Twido

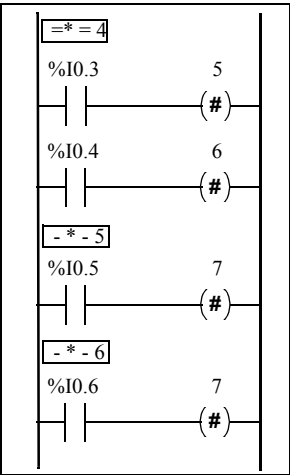


Programme liste  
d'instructions Twido

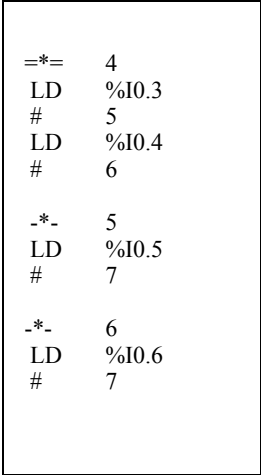
Séquence de divergences :



Non pris en charge

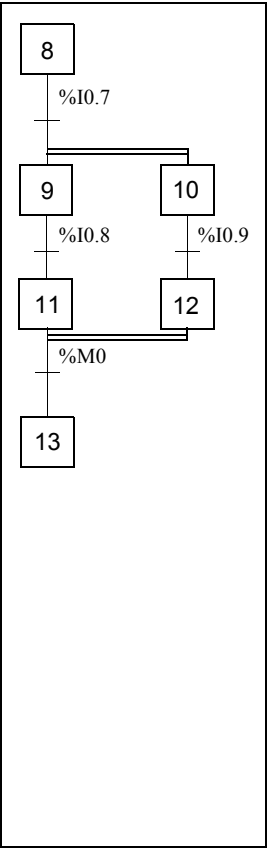


Programme schéma  
à contacts Twido

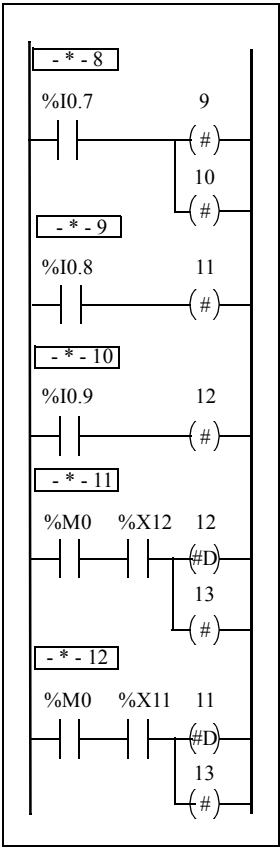


Programme liste  
d'instructions Twido

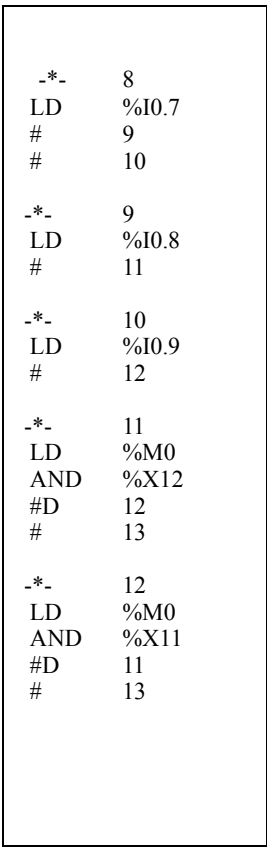
Séquences simultanées :



Non pris en charge



Programme schéma  
à contacts Twido



Programme liste  
d'instructions Twido

**Note :** Pour qu'un graphe Grafcet soit opérationnel, au moins une étape active doit être déclarée à l'aide de l'instruction \*=i (étape initiale) ou le graphe doit être prépositionné lors du pré-traitement à l'aide du bit système %S23 et de l'instruction S %Xi.



## Description de la structure d'un programme Grafcet

### Introduction

Un programme TwidoSoft Grafcet se déroule en trois phases :

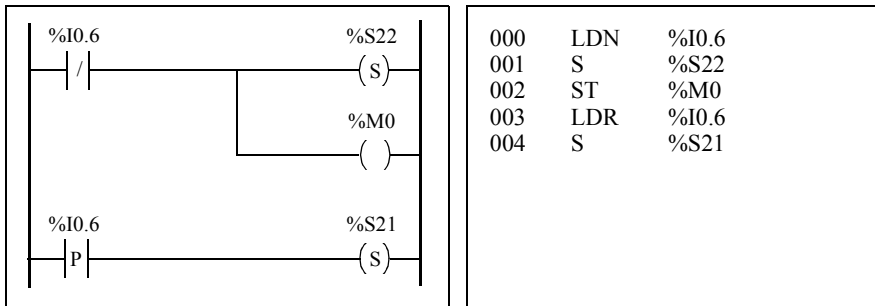
- Pré-traitement
- Traitement séquentiel
- Traitement postérieur

### Pré-traitement

Le pré-traitement gère les éléments suivants :

- les reprises de l'alimentation ;
- les défauts ;
- les changements du mode de fonctionnement ;
- le pré-positionnement des étapes Grafcet ;
- Logique d'entrée

Le front montant de l'entrée %I0.6 met à 1 le bit %S21. Cela a pour effet de désactiver les étapes actives et d'activer les étapes initiales.



Le pré-traitement commence à la première ligne du programme et se termine à la première occurrence d'une instruction "=" ou "-".

Trois bits système sont dédiés au contrôle du Grafcet : %S21, %S22 et %S23. Chaque bit système est mis à 1 (si nécessaire) par l'application, lors du pré-traitement généralement. La fonction associée est exécutée par le système à la fin du pré-traitement et le bit système est remis à 0 par le système.

Bit système	Nom	Description
%S21	Initialisation du Grafcet	Toutes les étapes actives sont désactivées et les étapes initiales sont activées.
%S22	Réinitialisation du Grafcet	Toutes les étapes sont désactivées.

Bit système	Nom	Description
%S23	Prépositionnement du Grafcet	Ce bit doit être mis à 1 si les objets %Xi sont explicitement écrits par l'application lors du pré-traitement. Si ce bit est maintenu sur 1 lors du pré-traitement sans changement explicite des objets %Xi, le Grafcet est figé (aucune mise à jour n'est prise en compte).

Traitement séquentiel

Le traitement séquentiel est exécuté dans le graphe (instructions représentant le graphe) :

- étapes
- actions associées aux étapes
- transitions
- conditions de transition

Exemple :

== 1

%I0.2

%I0.3

2

(#)

%I0.3

%I0.2

3

(#)

-\*- 2

%I0.4

1

(#)

-\*- 3

%I0.5

1

(#)

005 == 1

006 LD %I0.2

007 ANDN %I0.3

008 # 2

009 LD %I0.3

010 ANDN %I0.2

011 # 3

012 -\*- 2

013 LD %I0.4

014 # 1

015 -\*- 3

016 LD %I0.5

017 # 1

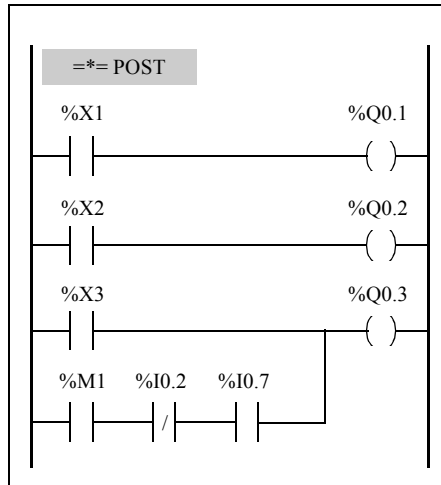
Le traitement séquentiel se termine par l'exécution de l'instruction "= \* = POST" ou par la fin du programme.

## Traitement postérieur

Le traitement postérieur gère les éléments suivants :

- les commandes du traitement séquentiel pour la régulation des sorties ;
- le verrouillage de sécurité spécifique aux sorties.

Exemple :



```

018  ==  POST
019  LD   %X1
020  ST   %Q0.1
021  LD   %X2
022  ST   %Q0.2
023  LD   %X3
024  OR(   %M1
025  ANDN  %I0.2
026  AND   %I0.7
027  )
028  ST   %Q0.3

```

## Actions associées aux étapes Grafcet

### Introduction

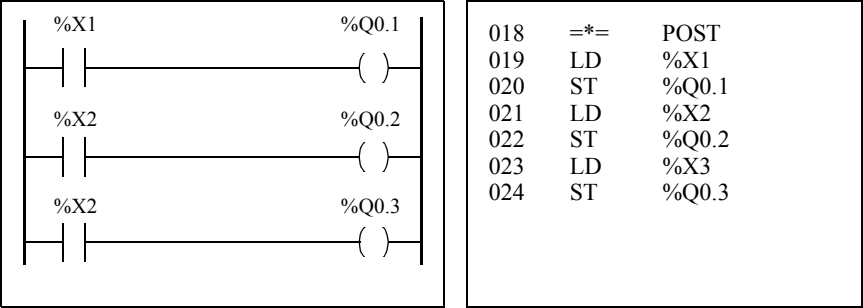
Un programme Grafcet TwidoSoft offre deux modes de programmation des actions associées aux étapes :

- dans la section de traitement postérieur ;
- dans les listes d'instructions ou les réseaux schéma à contacts des étapes mêmes.

### Association des actions dans le traitement postérieur

Si des contraintes de sécurité ou de mode d'exécution sont appliquées, il est préférable de programmer les actions dans la section de traitement postérieur d'une application Grafcet. Vous pouvez utiliser les instructions en langage liste d'instructions SET et RESET ou activer les bobines d'un programme schéma à contacts pour lancer les étapes Grafcet (%Xi).

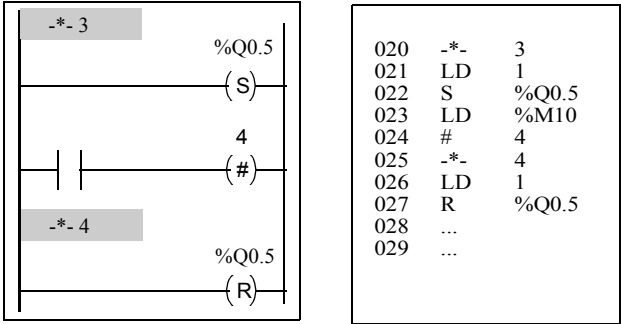
**Exemple :**



**Association  
d'actions à partir  
d'une application**

Vous pouvez programmer les actions associées aux étapes sous forme de listes d'instructions ou de réseaux schéma à contacts. Dans ce cas, la liste d'instructions ou le réseau schéma à contacts n'est pas scruté(e), tant que l'étape n'est pas active. Ce mode d'utilisation du langage Grafcet est le plus efficace, le plus lisible et le plus facile à gérer.

**Exemple :**





---

## Description des instructions et des fonctions

# IV

---

### Présentation

#### Objet de cette partie

Cette partie fournit des descriptions détaillées des instructions élémentaires et avancées, ainsi que des bits et des mots système des langages Twido.

#### Contenu de cette partie

Cette partie contient les chapitres suivants :

Chapitre	Titre du chapitre	Page
14	Instructions élémentaires	313
15	Instructions avancées	385
16	Bits système et mots système	545





---

## Instructions élémentaires

# 14

---

### Présentation

#### Objet de ce chapitre

Ce chapitre fournit des détails sur les instructions et les blocs fonctions utilisés pour créer des programmes de régulation élémentaires des automates Twido.

#### Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
14.1	Traitement booléen	315
14.2	Blocs fonctions élémentaires	332
14.3	Traitement numérique	358
14.4	Instructions sur programme	377

---



---

## 14.1            Traitement booléen

---

### Présentation

**Objet de ce sous-chapitre**    Ce sous-chapitre offre une introduction au traitement booléen. Elle s'appuie sur des descriptions et des directives de programmation d'instructions booléennes.

---

**Contenu de ce sous-chapitre**    Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Instructions booléennes	316
Explication du format de description des instructions booléennes	318
Instructions de chargement (LD, LDN, LDR, LDF)	320
Instructions d'affectation (ST, STN, R, S)	322
Instructions AND logique (AND, ANDN, ANDR, ANDF)	324
Instructions OR logique (OR, ORN, ORR, ORF)	326
OR exclusif, instructions (XOR, XORN, XORR, XORF)	328
Instruction NOT (N)	330

---

## Instructions booléennes

### Introduction

Les instructions booléennes s'apparentent aux éléments graphiques du langage schéma à contacts. Ces instructions sont présentées dans le tableau suivant.

Élément	Instruction	Exemple	Description
Éléments de test	L'instruction de chargement (LD) équivaut à un contact ouvert.	LD %I0.0	Le contact est fermé lorsque le bit %I0.0 se trouve à l'état 1.
Éléments d'action	L'instruction de stockage (ST) équivaut à une bobine.	ST %Q0.0	L'objet bit associé prend la valeur logique de l'accumulateur de bit (résultat de la logique précédente).

Le résultat booléen des éléments de test est appliqué aux éléments d'action, comme l'illustrent les instructions suivantes.

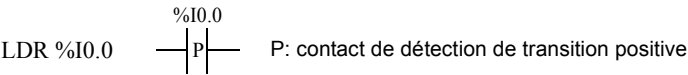
LD    %I0.0  
AND  %i0.1  
ST    %Q0.0

### Test des entrées de l'automate

Des instructions sur test booléennes peuvent être utilisées pour détecter des fronts montants ou descendants sur les entrées de l'automate. Un front est détecté lorsque l'état d'une entrée est passé de la valeur "scrutation n-1" à la valeur "scrutation n" courante. La détection de ce front reste effective pendant la scrutation courante.

### Front montant, détection

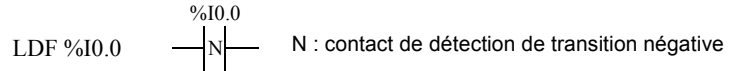
L'instruction LDR (Load Rising Edge - chargement du front montant) équivaut à un contact de détection d'un front montant. Le front montant détecte le passage de la valeur d'entrée de 0 à 1.  
Un contact de détection de transition positive est utilisé pour détecter un front montant, comme l'illustre le schéma suivant.




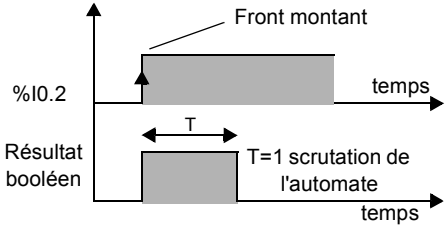

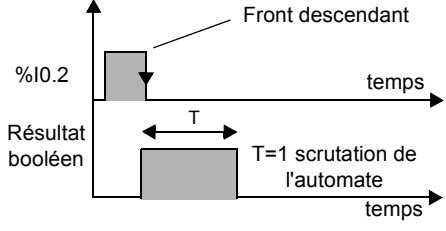
**Détection d'un front descendant**

L'instruction LDF (Load Falling Edge - chargement du front descendant) équivaut à un contact de détection d'un front descendant. Le front descendant détecte le passage de la valeur d'entrée de régulation de 1 à 0.

Un contact de détection de transition négative est utilisé pour détecter un front descendant, comme l'illustre le schéma suivant.

**Détection d'un front**

Le tableau suivant résume les instructions de détection de fronts et leurs chronogramme :

Front	Instruction sur test	Schéma à contacts	Chronogramme
Front montant	LDR %I0.0		
Front descendant	LDF %I0.0		

**Note** : il est dorénavant possible d'appliquer les instructions sur front aux bits internes %Mi.

## Explication du format de description des instructions booléennes

### Introduction

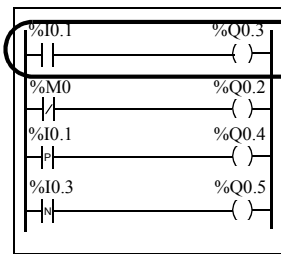
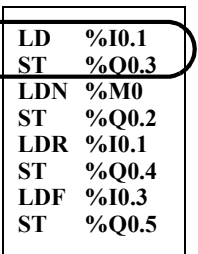
Chaque instruction booléenne de ce sous-chapitre est décrite à l'aide des informations suivantes :

- Description rapide
- Exemple représentant l'instruction et le schéma à contacts correspondant
- Liste d'opérandes autorisés
- Chronogramme

Les explications ci-dessous présentent plus en détails le mode de description des instructions booléennes de ce sous-chapitre.

### Exemples

L'illustration suivante présente le mode d'affichage des exemples pour chaque instruction.

	
---	--

Equivalents dans le langage  
schéma à contacts

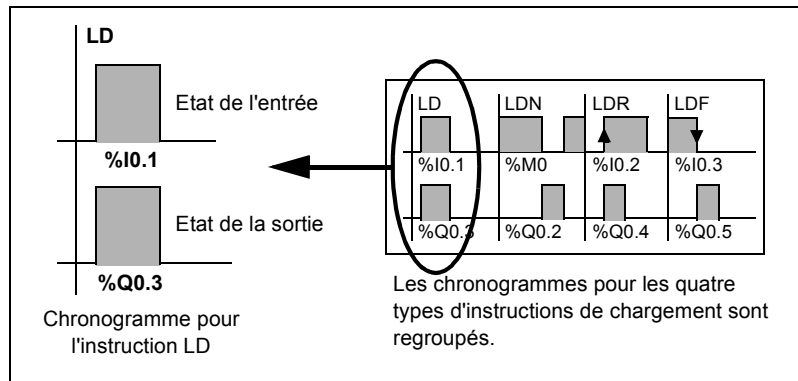
Instructions en langage  
liste d'instructions

### Opérandes autorisés

Le tableau suivant définit les types d'opérandes autorisés et utilisés dans les instructions booléennes.

Opérande	Description
O/1	Valeur immédiate de 0 ou 1
%I	Entrée automate %Ii.j
%Q	Sortie automate %Qi.j
%M	Bit interne %Mi
%S	Bit système %Si
%X	Bit étape %Xi
%BLK.x	Bit bloc fonction (%Tmi.Q, par exemple)
%•Xk	Bit mot (%MWi:Xk, par exemple)
[	Expression de comparaison ([%MWi<1000], par exemple)

**Chronogrammes** L'illustration suivante présente le mode d'affichage des chronogrammes pour chaque instruction.



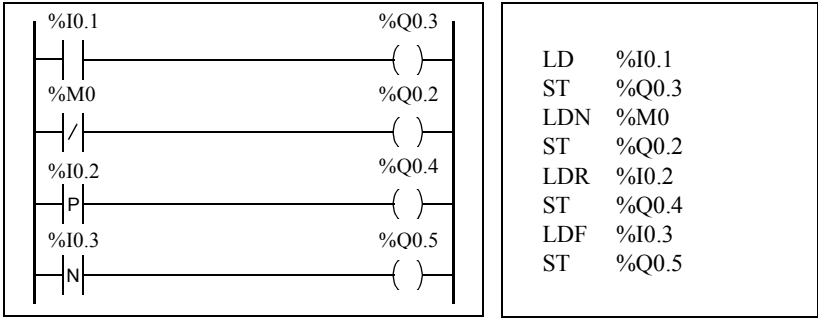
## Instructions de chargement (LD, LDN, LDR, LDF)

### Introduction

Les instructions de chargement LD, LDN, LDR et LDF correspondent respectivement aux contacts Ouvert, Fermé, Front montant et Front descendant (les instructions LDR et LDF ne sont utilisées qu'avec des entrées et des mots internes de l'automate et des entrées d'esclaves AS-Interface).

### Exemples

Les schémas suivants sont des exemples d'instructions de chargement.



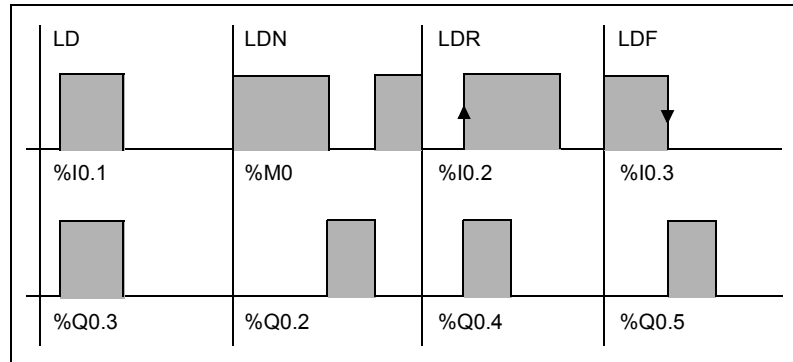
### Opérandes autorisés

Le tableau suivant répertorie les types d'instructions de chargement, leurs équivalents dans le langage schéma à contacts, ainsi que les opérandes autorisés.

Instruction en langage liste	Symbole équivalent dans un schéma à contacts	Opérandes autorisés
LD		0/1, %I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %*:Xk,[
LDN		0/1, %I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %*:Xk,[
LDR		%I, %IA, %M
LDF		%I, %IA, %M



**Chronogramme** L'illustration suivante montre le chronogramme des instructions de chargement :



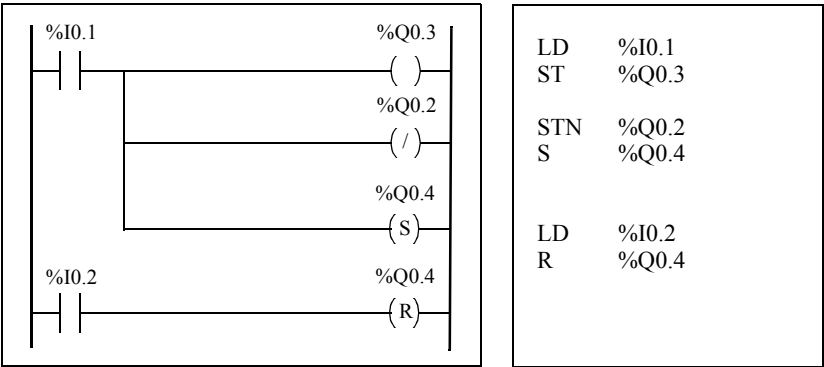
Instructions d’affectation (ST, STN, R, S)

Introduction

Les instructions d’affectation ST, STN, S et R correspondent respectivement aux bobines directes, inverses, d’enclenchement et de déclenchement.

Exemples

Les schémas suivants sont des exemples d’instructions d’affectation.

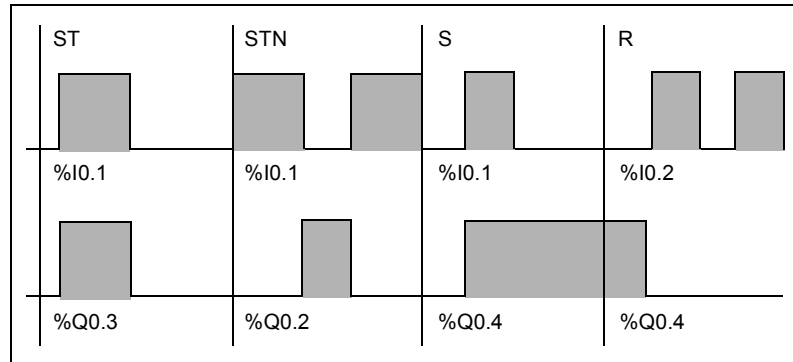


Opérandes autorisés

Le tableau suivant répertorie les types d’instructions d’affectation, leurs équivalents dans le langage schéma à contacts, ainsi que les opérandes autorisés.

Instruction en langage liste	Symbole équivalent dans un schéma à contacts	Opérandes autorisés
ST	( )	%Q,%QA,%M,%S,%BLK.x,%•:Xk
STN	( / )	%Q,%QA%M,%S,%BLK.x,%•:Xk
S	( S )	%Q,%QA,%M,%S,%X,%BLK.x,%•:Xk
R	( R )	%Q,%QA,%M,%S,%X,%BLK.x,%•:Xk

**Chronogramme** L'illustration suivante montre le chronogramme des instructions d'affectation.



Instructions AND logique (AND, ANDN, ANDR, ANDF)

Introduction

Les instructions AND effectuent une opération de liaison AND logique entre l'opérande (ou son inverse, ou son front montant ou descendant) et le résultat booléen de l'instruction précédente.

Exemples

Les schémas suivants sont des exemples d'instructions AND.

%I0.1

%M1

%Q0.3

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

(

)

%M2

%I0.2

%Q0.2

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

(

)

%I0.3

%I0.4

%Q0.4

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

(

S

)

%M3

%I0.5

%Q0.5

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

|

(

S

)

LD

%I0.1

AND

%M1

ST

%Q0.3

LD

%M2

ANDN

%I0.2

ST

%Q0.2

LD

%I0.3

ANDR

%I0.4

S

%Q0.4

LD

%M3

ANDF

%I0.5

S

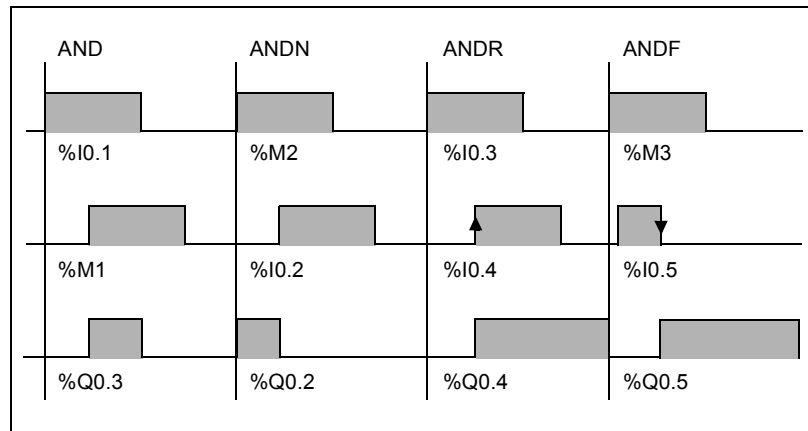
%Q0.5

Opérandes autorisés

Le tableau suivant répertorie les types d'instructions AND, leurs équivalents dans le langage schéma à contacts, ainsi que les opérandes autorisés.

Instruction en langage liste	Symbole équivalent dans un schéma à contacts	Opérandes autorisés
AND	<div><div><div> </div><div> </div><div> </div></div></div>	0/1, %I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk, [
ANDN	<div><div><div> </div><div> </div><div> </div><div> </div></div></div>	0/1, %I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk, [
ANDR	<div><div><div> </div><div> </div><div> </div><div> </div><div>P</div></div></div>	%I, %IA, %M
ANDF	<div><div><div> </div><div> </div><div> </div><div> </div><div>N</div></div></div>	%I, %IA, %M

**Chronogramme** L'illustration suivante montre le chronogramme des instructions AND.



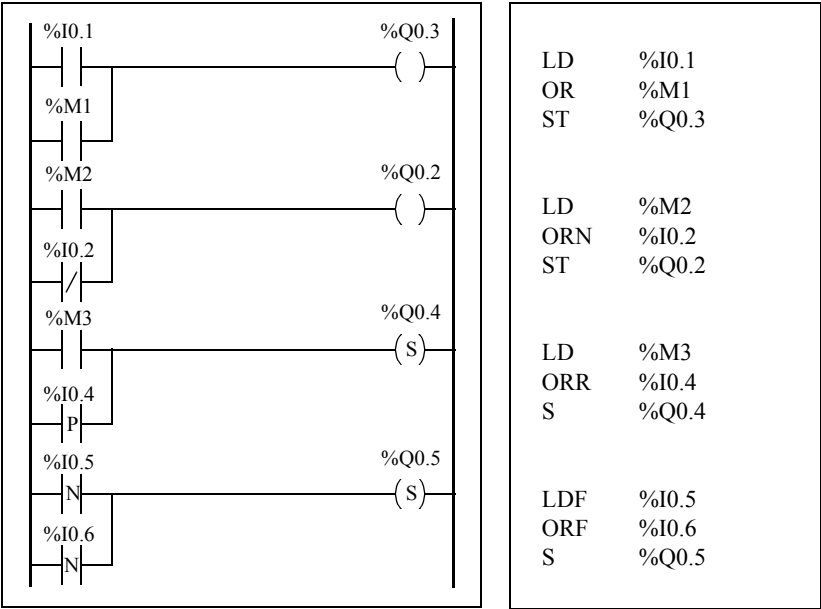
## Instructions OR logique (OR, ORN, ORR, ORF)

### Introduction

Les instructions OR effectuent une opération de liaison OR logique entre l'opérande (ou son inverse, ou son front montant ou descendant) et le résultat booléen de l'instruction précédente.

### Exemples

Les schémas suivants sont des exemples d'instructions OR.



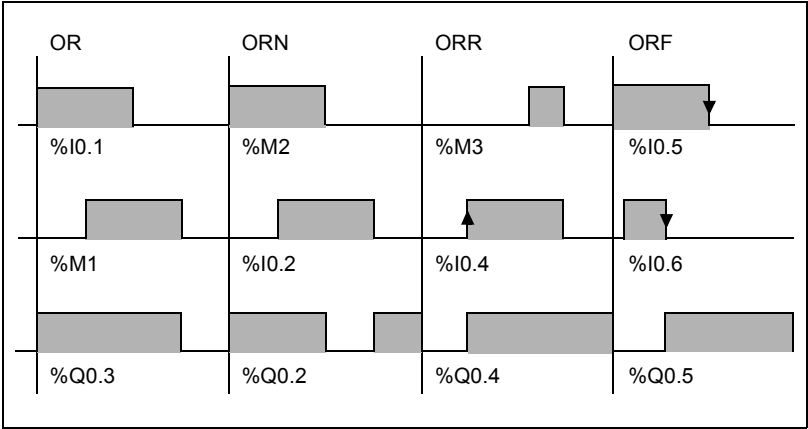
Opérandes  
autorisés

Le tableau suivant répertorie les types d'instructions OR, leurs équivalents dans le langage schéma à contacts, ainsi que les opérandes autorisés.

Instruction en langage liste	Symbole équivalent dans un schéma à contacts	Opérandes autorisés
OR		0/1, %I,%IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk
ORN		0/1, %I,%IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk
ORR		%I, %IA, %M
ORF		%I, %IA, %M

Chronogramme

L'illustration suivante montre le chronogramme des instructions OR.



OR exclusif, instructions (XOR, XORN, XORR, XORF)

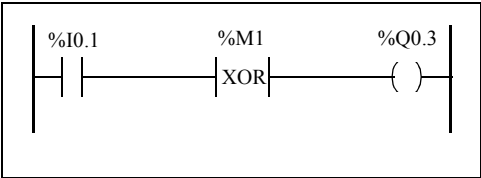
Introduction

Les instructions XOR effectuent une opération de liaison OR exclusif entre l'opérande (ou son inverse, ou son front montant ou descendant) et le résultat booléen de l'instruction précédente.

Exemples

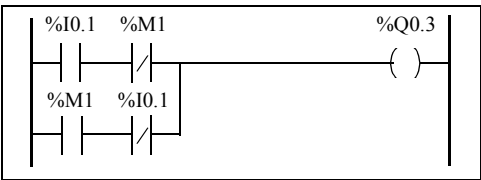
L'exemple suivant illustre l'utilisation d'instructions XOR.

Schéma avec instruction XOR :



```
LD    %I0.1
XOR   %M1
ST    %Q0.3
```

Schéma équivalent sans instruction XOR :



```
LD    %I0.1
ANDN  %M1
OR(    %M1
ANDN  %I0.1
)
ST    %Q0.3
```

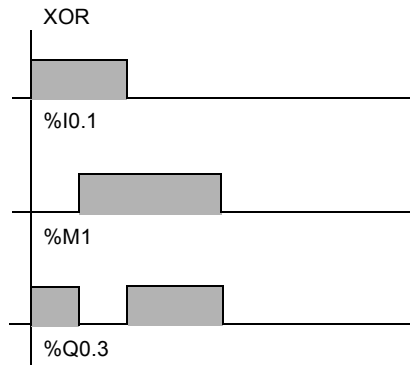
Opérandes autorisés

Le tableau suivant répertorie les types d'instructions XOR, ainsi que les opérandes autorisés.

Instruction langage liste	Opérandes autorisés
XOR	%I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %*:Xk
XORN	%I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %*:Xk
XORR	%I, %IA, %M
XORF	%I, %IA, %M



**Chronogramme** Le diagramme suivant illustre la temporisation des instructions XOR.

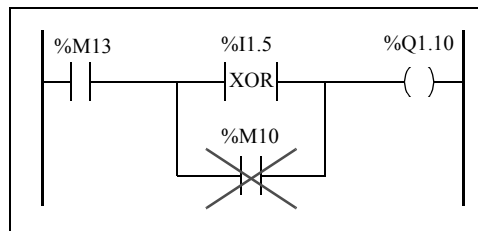


### Cas spécifiques

Veuillez observer les précautions suivantes lors de l'utilisation d'instructions XOR dans des programmes en langage schéma à contacts :

- Ne commencez jamais un réseau par un contact XOR.
- N'insérez jamais de contacts XOR parallèlement à d'autres éléments du schéma à contacts (reportez-vous à l'exemple suivant.)

Comme l'illustre l'exemple suivant, l'insertion d'un élément parallèle à un contact XOR générera une erreur de validation.



## Instruction NOT (N)

---

**Introduction** L'instruction NOT (N) inverse le résultat booléen de l'instruction précédente.

---

**Exemple** L'exemple suivant illustre l'utilisation de l'instruction NOT.

LD	%I0.1
OR	%M2
ST	%Q0.2
N	
AND	%M3
ST	%Q0.3

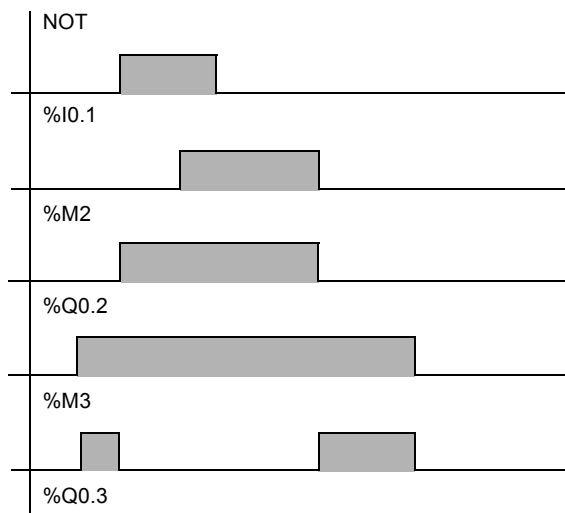
<b>Note :</b> L'instruction NOT n'est pas réversible.
---

---

**Opérandes autorisées** Sans objet.

---

**Chronogramme** L'illustration suivante montre le chronogramme de l'instruction NOT.



## 14.2            Blocs fonctions élémentaires

---

### Présentation

**Objet de ce sous-chapitre**    Ce sous-chapitre présente des descriptions et des conseils de programmation relatifs aux blocs fonctions élémentaires.

---

**Contenu de ce sous-chapitre**    Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Blocs fonctions standards	333
Principes de programmation de blocs fonction standards	335
Bloc fonction temporisateur (%TMI)	337
Type de temporisateur TOF	339
Type de temporisateur TON	340
Type de temporisateur TP	341
Programmation et configuration de temporisateurs	342
Bloc fonction compteur/décompteur (%Ci)	345
Programmation et configuration des compteurs	349
Bloc fonction registre bits à décalage (%SB Ri)	351
Bloc fonction pas à pas (%SCi)	354

---

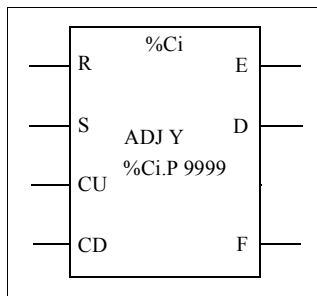
## Blocs fonctions standards

### Introduction

Les blocs fonctions sont les sources des objets bits et des mots spécifiques utilisés par les programmes. Les blocs fonctions élémentaires comportent des fonctions simples telles que des temporisateurs ou des compteurs/décompteurs.

### Exemple de bloc fonction

L'illustration suivante présente un exemple de bloc fonction compteur/décompteur.



Bloc compteur/décompteur

### Objets bits

Les objets bits correspondent aux sorties des blocs. Les instructions booléennes de test peuvent accéder à ces bits selon l'une ou l'autre de ces méthodes :

- directement (LD E, par exemple) s'ils sont liés au bloc par une programmation réversible (voir rubrique *Principes de programmation de blocs fonction standards*, p. 335).
- en spécifiant le type de bloc (LD %Ci.E, par exemple).

Les entrées sont accessibles sous forme d'instructions.

### Objets mots

Les objets mots correspondent aux :

- **Paramètres de configuration du bloc** : Le programme peut accéder à certains paramètres (paramètres de pré-sélection, par exemple), mais pas à d'autres (base temps, par exemple).
- **Valeurs courantes** : %Ci.V, la valeur de comptage courante, par exemple.

**Objets bits et  
objets mots  
accessibles**

Le tableau suivant décrit les objets bits et les objets mots de blocs fonctions auxquels le programme a accès.

Bloc fonction standard	Symbole	Plage (i)	Types d'objets	Description	Adresse	Accès en mode écriture
Temporisateur	%T <i>Mi</i>	0 - 127	Mot	Valeur courante	%T <i>Mi</i> .V	non
				Valeur de présélection	%T <i>Mi</i> .P	oui
			Bit	Sortie du temporisateur	%T <i>Mi</i> .Q	non
Compteur/ Décompteur	%C <i>i</i>	0 - 127	Mot	Valeur courante	%C <i>i</i> .V	non
				Valeur de présélection	%C <i>i</i> .P	oui
			Bit	Sortie pour dépassement par valeur inférieure (vide)	%C <i>i</i> .E	non
				Sortie prédéfinie atteinte	%C <i>i</i> .D	non
				Sortie pour débordement (plein)	%C <i>i</i> .F	non

## Principes de programmation de blocs fonction standards

### Introduction

Pour programmer des blocs fonction standards, appliquez l'une des méthodes suivantes :

- Instructions sur bloc fonction (par exemple `BLK %TM2`) : Cette méthode de programmation en langage schéma à contacts réversible permet l'exécution d'opérations sur le bloc, à un emplacement unique du programme.
- Instructions spécifiques (par exemple `CU %Ci`) : Cette méthode non réversible permet l'exécution d'opérations sur les entrées du bloc, à plusieurs emplacements du programme (par exemple, line 100 `CU %C1`, line 174 `CD %C1`, line 209 `LD %C1.D`).

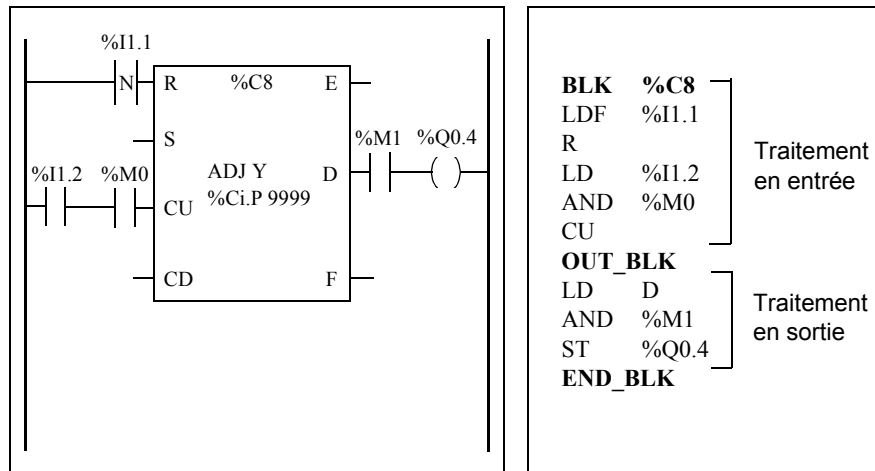
### Programmation réversible

Utilisez les instructions `BLK`, `OUT_BLK` et `END_BLK` pour une programmation réversible :

- **BLK** : Indique le début du bloc.
- **OUT\_BLK** : Utilisé pour câbler directement les sorties du bloc.
- **END\_BLK** : Indique la fin du bloc.

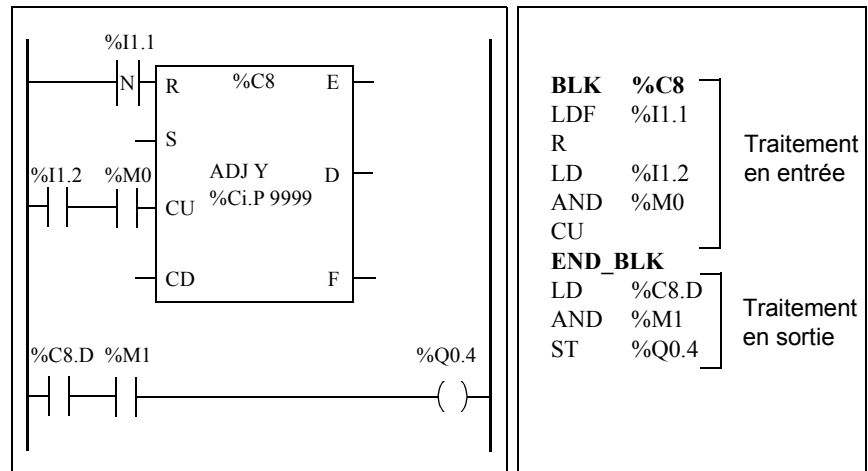
### Exemple avec des sorties câblées

Vous trouverez ci-dessous un exemple de programmation réversible d'un bloc fonction compteur avec des sorties câblées.



### Exemple sans sortie câblée

Vous trouverez ci-dessous un exemple de programmation réversible d'un bloc fonction compteur dépourvu de sortie câblée.



**Note :** Seules les instructions de test et d'entrée sur le bloc correspondant peuvent être placées entre les instructions BLK et OUT\_BLK (ou entre BLK et END\_BLK lorsque OUT\_BLK n'est pas programmé).



## Bloc fonction temporisateur (%TMi)

### Introduction

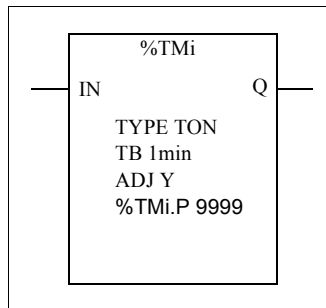
Il existe trois types de blocs fonction temporisateur :

- TON (temporisateur de retard à l'enclenchement) : ce type de temporisateur permet de gérer les retards à l'enclenchement.
- TOF (temporisateur de retard au déclenchement) : ce type de temporisateur permet de gérer les retards au déclenchement.
- TP (temporisateur - Impulsion) : ce type de temporisateur permet de générer des impulsions d'une durée précise.

TwidoSoft permet de programmer et de modifier les retards de ces temporisateurs et/ou les durées des impulsions qu'ils génèrent.

### Illustration

L'exemple suivant illustre l'utilisation du bloc fonction temporisateur.



Bloc fonction temporisateur

**Paramètres**

Le bloc fonction temporisateur possède les paramètres suivants :

Paramètre	Etiquette	Valeur
Numéro du temporisateur	%Tmi	0 à 63 : TWDLCAA10DRF et TWDLCAA16DRF 0 à 127 pour tous les autres automates.
Type	TON	• retard à l'enclenchement (par défaut)
	TOF	• retard au déclenchement
	TP	• impulsion (monostable)
Base de temps	TB	1 min (par défaut), 1 s, 100 ms, 10 ms, 1 ms
Valeur courante	%Tmi.V	Mot avec des incréments allant de 0 à %Tmi.P lorsque le temporisateur est en cours d'exécution. Peut être lu et testé, mais pas écrit par le programme. %Tmi.V peut être modifié par l'éditeur de tables d'animation.
Valeur de présélection	%Tmi.P	0 - 9999. Mot pouvant être lu, testé et écrit par le programme. La valeur par défaut est 9999. La période ou le délai généré est égal à %Tmi.P x TB.
Editeur de tables d'animation	Y/N	Y : Oui, la valeur %Tmi.P de présélection peut être modifiée à l'aide de l'éditeur de tables d'animation. N : Non, la valeur %Tmi.P de présélection ne peut pas être modifiée.
Entrée validation (ou de l'instruction)	IN	Démarre le temporisateur sur le front montant (types TON ou TP) ou descendant (type TOF).
Sortie du temporisateur	Q	Le bit associé %Tmi.Q est réglé sur 1 en fonction de la fonction exécutée : TON, TOF ou TP

**Note :** Plus la valeur de présélection est grande, plus le temporisateur sera précis.

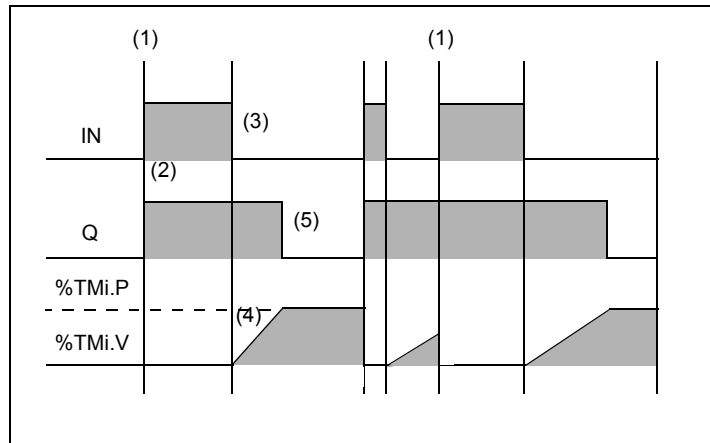
## Type de temporisateur TOF

### Introduction

Le type de temporisateur TOF (Timer Off-Delay, temporisateur à retard de déclenchement) permet de gérer des retards au déclenchement. TwidoSoft permet de programmer ce retard.

### Chronogramme

Le chronogramme suivant illustre le fonctionnement du type de temporisateur TOF.



### Fonctionnement

Le tableau suivant décrit le fonctionnement du type de temporisateur TOF.

Phase	Description
1	La valeur courante %TMI.V prend la valeur 0 sur un front montant en entrée IN, et ce, même si le temporisateur est en cours d'exécution.
2	Le bit de sortie %TMI.Q passe à 1 lorsqu'un front montant est détecté en entrée IN.
3	Le temporisateur démarre sur le front descendant de l'entrée IN.
4	La valeur courante %TMI.V augmente jusqu'à %TMI.P, par incréments d'une unité à chaque pulsation de la base temps TB.
5	Le bit de sortie %TMI.Q est remis à 0 lorsque la valeur courante atteint %TMI.P.

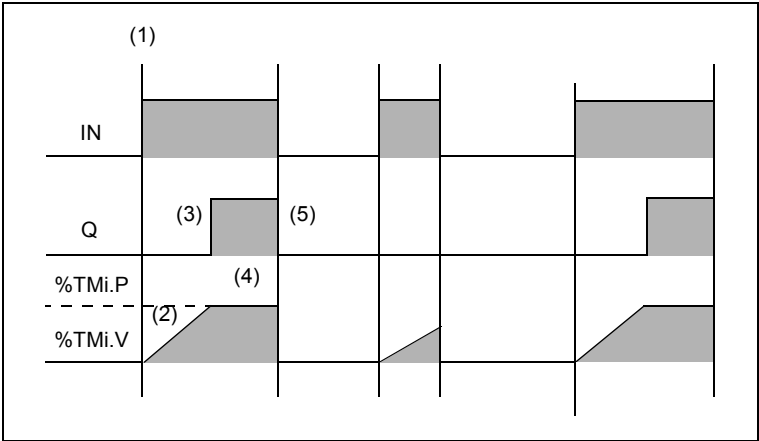
## Type de temporisateur TON

### Introduction

Le type de temporisateur TON (Timer On-Delay, temporisateur à retard à l'enclenchement) permet de gérer des retards à l'enclenchement. TwidoSoft permet de programmer ce retard.

### Chronogramme

Le chronogramme suivant illustre le fonctionnement du type de temporisateur TON.



### Fonctionnement

Le tableau suivant décrit le fonctionnement du type de temporisateur TON.

Phase	Description
1	Le temporisateur démarre sur le front montant de l'entrée IN.
2	La valeur courante %Tmi.V augmente de 0 à %Tmi.P, par incréments d'une unité à chaque pulsation de la base temps TB.
3	Le bit de sortie %Tmi.Q passe à 1 lorsque la valeur courante a atteint %Tmi.P.
4	Le bit de sortie %Tmi.Q conserve la valeur 1 tant que la valeur de l'entrée IN est à 1.
5	Lorsqu'un front descendant est détecté en entrée IN, le temporisateur s'arrête, et ce, même s'il n'a pas atteint %Tmi.P et que %Tmi.V est réglé sur 0.

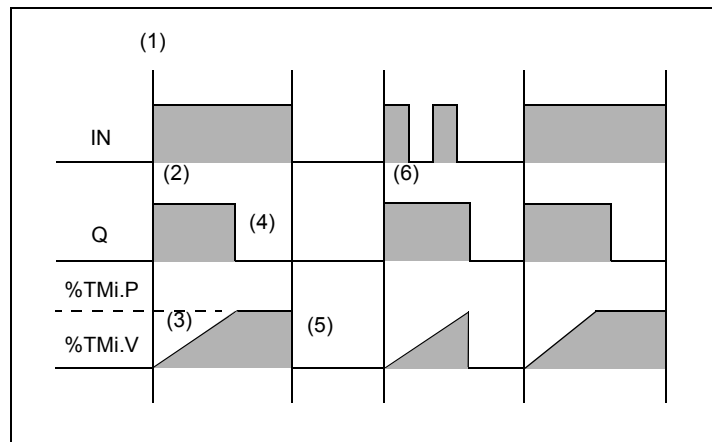
## Type de temporisateur TP

### Introduction

Le type de temporisateur TP (Timer – Pulse, Temporisateur – Impulsion) permet de générer des impulsions d'une durée spécifique. TwidoSoft permet de programmer cette durée.

### Chronogramme

Le chronogramme suivant illustre le fonctionnement du type de temporisateur TP.



### Fonctionnement

Le tableau suivant décrit le fonctionnement du type de temporisateur TP.

Phase	Description
1	Le temporisateur démarre sur le front montant de l'entrée IN. La valeur courante %Tmi.V est mis à 0 si le temporisateur n'a pas encore démarré.
2	Le bit de sortie %Tmi.Q est mis à 1 lorsque le temporisateur démarre.
3	La valeur courante %Tmi.V du temporisateur augmente de 0 à %Tmi.P, par incréments d'une unité à chaque pulsation de la base temps TB.
4	Le bit de sortie %Tmi.Q est mis à 0 lorsque la valeur courante atteint %Tmi.P.
5	La valeur courante %Tmi.V est mis à 0 lorsque %Tmi.V égale %Tmi.P et que l'entrée IN retrouve la valeur 0.
6	Le temporisateur ne peut pas être remis à zéro. Lorsque %Tmi.V égale %Tmi.P et que l'entrée IN est mis à 0, %Tmi.V est réglé sur 0.

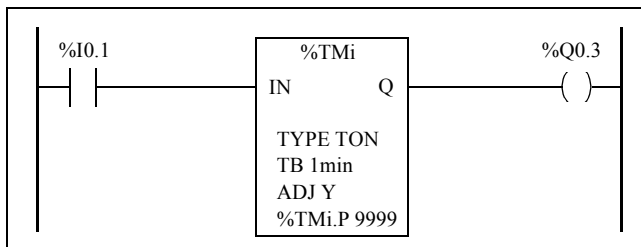
## Programmation et configuration de temporisateurs

### Introduction

Tous les blocs fonction temporisateur (%TMi) sont programmés de la même façon, indépendamment de leur mode d'utilisation. La fonction temporisateur (TON, TOF ou TP) est sélectionnée au moment de la configuration.

### Exemples

L'illustration suivante représente un bloc fonction temporisateur et affiche des exemples de programmation réversible et non réversible.



#### Programmation réversible

```

BLK   %TM1
LD     %I0.1
IN
OUT_BLK
LD     Q
ST     %Q0.3
END_BLK

```

#### Programmation non réversible

```

LD     %I0.1
IN     %TM1
LD     %TM1.Q
ST     %Q0.3

```

### Configuration

Les paramètres suivants doivent être saisis au moment de la configuration :

- Type de temporisateur : TON, TOF ou TP
- Base temps (TB) : 1 min, 1 s, 100 ms, 10 ms ou 1 ms
- Valeur de présélection (%TMi.P) : 0 à 9 999
- Réglage : coché ou non coché

**Cas particuliers**

Le tableau suivant présente une liste des cas spécifiques de programmation du bloc fonction temporisateur.

Cas spécifique	Description
Effet d'un redémarrage à froid (%S0=1)	Force la valeur courante sur 0. Règle la sortie %TMi.Q sur 0. La valeur de présélection reprend la valeur réglée au moment de la configuration.
Effet d'une reprise à chaud (%S1=1)	N'a aucun effet sur la valeur courante et la valeur de présélection du temporisateur. La valeur courante n'est pas modifiée lors d'une coupure d'alimentation secteur.
Effet d'un arrêt de l'automate	L'arrêt de l'automate ne provoque pas le gel de la valeur courante.
Effet d'un saut de programme	Le saut d'un bloc temporisateur ne provoque pas le gel du temporisateur. L'incréméntation du temporisateur se poursuit jusqu'à ce que la valeur de présélection (%TMi.P) soit atteinte. A ce stade, l'état du bit Terminé (%TMi.Q) affecté à la sortie Q du bloc temporisateur est modifié. Cependant, la sortie associée, liée directement à la sortie du bloc, n'est ni activée, ni scrutée par l'automate.
Test par bit %TMi.Q (bit terminé)	Nous conseillons de ne tester le bit %TMi.Q qu'une seule fois dans le programme.
Effet de la modification de la valeur de présélection de %TMi.P	La modification de la valeur de présélection à l'aide d'une instruction ou d'un réglage ne prend effet qu'à la prochaine activation du temporisateur.

**Temporisateurs avec base temps de 1 ms**

La base temps de 1 ms n'est disponible qu'avec les cinq premiers temporisateurs. Les quatre mots système %SW76, %SW77, %SW78 et SW79 peuvent être utilisés comme des "sabliers". Ces quatre mots sont décréémentés de manière individuelle par le système toutes les millisecondes, **si leur valeur est positive**.

Il est possible de créer une temporisation multiple en chargeant successivement un de ces mots ou en testant les valeurs intermédiaires. Les valeurs négatives de ces quatre mots ne seront pas modifiées. Un temporisateur peut être "gelé" en réglant le bit 15 sur la valeur 1, puis "dégelé" en remettant à zéro cette valeur.

**Exemple de programmation**

L'exemple suivant illustre la programmation d'un bloc fonction temporisateur.

```
LDR    %I0.1      (Lancement du temporisateur sur le front montant de  
%I0.1)  
[%SW76:=XXXX]    (XXXX = valeur requise)  
LD      %I0.2      (gestion optionnelle du gel, gel de l'entrée I0.2)  
ST      %SW76:X15  
LD      [%SW76=0]   (test du temporisateur de fin)  
ST      %M0  
.....
```





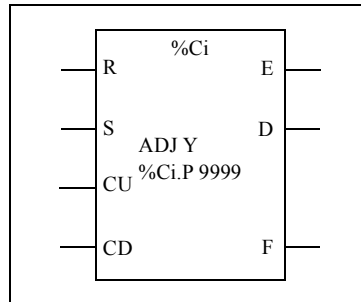
## Bloc fonction compteur/décompteur (%Ci)

### Introduction

Le bloc fonction compteur (%Ci) permet de compter ou de décompter des événements. Ces deux opérations peuvent être réalisées simultanément.

### Illustration

L'illustration suivante présente un exemple de bloc fonction compteur/décompteur.



Bloc fonction compteur/décompteur

**Paramètres**

Le bloc fonction compteur possède les paramètres suivants :

Paramètre	Etiquette	Valeur
Numéro du compteur	%Ci	0 à 127
Valeur courante	%Ci.V	La valeur du mot est augmentée ou diminuée d'une unité en fonction des entrées (ou des instructions) CU et CD. Peut être lue et testée, mais pas écrite par le programme. Utilisez l'éditeur de données pour modifier %Ci.V.
Valeur de présélection	%Ci.P	$0 \leq \%Ci.P \leq 9999$ . Le mot peut être lu, testé et écrit (valeur par défaut : 9999).
Edition à l'aide de l'Editeur de tables d'animation	ADJ	<ul style="list-style-type: none"> <li>● Y : Oui, la valeur de présélection peut être modifiée à l'aide de l'éditeur de tables d'animation.</li> <li>● N : Non, la valeur de présélection ne peut pas être modifiée à l'aide de l'éditeur de tables d'animation.</li> </ul>
Entrée (ou instruction) RAZ	R	A l'état 1 : %Ci.V = 0.
Entrée (ou instruction) de présélection	S	A l'état 1 : %Ci.V = %Ci.P.
Entrée (ou instruction) de comptage	CU	Augmente la valeur de %Ci.V d'une unité sur un front montant.
Entrée (ou instruction) de décomptage	CD	Diminue la valeur de %Ci.V d'une unité sur un front montant.
Sortie débordement décomptage	E (vide)	Le bit associé %Ci.E est égal à 1, lorsque la valeur du décompteur %Ci.V passe de 0 à 9999 (mis à 1 lorsque %Ci.V atteint 9999 et remis à zéro si le décomptage se poursuit).
Sortie prédéfinie atteinte	D (Terminé)	Le bit associé %Ci.D est égal à 1, lorsque %Ci.V est égal à %Ci.P.
Sortie débordement comptage	F (plein)	Le bit associé %Ci.F est égal à 1, lorsque la valeur de %Ci.V passe de 9999 à 0 (mis à 1 lorsque %Ci.V atteint 0 et remis à zéro si le comptage croissant se poursuit).

**Fonctionnement** Le tableau suivant décrit les étapes principales des opérations de comptage et de décomptage.

Fonctionnement	Action	Résultat
Comptage	Un front montant apparaît sur l'entrée comptage CU (ou l'instruction CU est activée).	La valeur courante de %Ci.V est augmentée d'une unité.
	La valeur courante de %Ci.V est égale à la valeur de présélection de %Ci.P.	Le bit de sortie "présélection atteinte" %Ci.D passe à 1.
	La valeur courante de %Ci.V passe de 9999 à 0.	Le bit de sortie %Ci.F (débordement comptage) passe à 1.
	Si le comptage se poursuit.	Le bit de sortie %Ci.F (débordement comptage) est remis à zéro.
Décomptage	Un front montant apparaît sur l'entrée décomptage CD (ou l'instruction CD est activée).	La valeur courante de %Ci.V est diminuée d'une unité.
	La valeur courante de %Ci.V passe de 0 à 9999.	Le bit de sortie %Ci.E (débordement décomptage) passe à 1.
	Si le décomptage se poursuit.	Le bit de sortie %Ci.F (débordement décomptage) est remis à zéro.
Comptage/ Décomptage	Pour utiliser simultanément les fonctions de comptage et de décomptage (ou pour activer les deux instructions CD et CU), les deux entrées CU et CD correspondantes doivent être commandées simultanément. Ces deux entrées sont ensuite scrutées. Si leur valeur est égale à 1, la valeur courante n'est pas modifiée.	
Remise à zéro	Mise à 1 de l'entrée R (ou l'instruction R est activée).	Force la remise à zéro de la valeur %Ci.V. Les sorties %Ci.E, %Ci.D et %Ci.F sont mises à 0. L'entrée remise à zéro est prioritaire.
Présélection	Si l'entrée S est mis à 1 (ou si l'instruction S est activée) et que l'entrée de remise à zéro est à l'état 0 (ou que l'instruction R est inactive).	La valeur courante %Ci.V prend la valeur de %Ci.P et la sortie %Ci.D est mis à 1.

**Cas spécifiques**

Le tableau suivant présente une liste de cas spécifiques de fonctionnement et de configuration des compteurs.

Cas spécifique	Description
Effet d'un redémarrage à froid (%S0=1)	<ul style="list-style-type: none"><li>● La valeur courante de %Ci.V est mise à 0.</li><li>● Les bits de sortie %Ci.E, %Ci.D et %Ci.F sont mis à sur 0.</li><li>● La valeur de présélection est initialisée avec la valeur définie au moment de la configuration</li></ul>
Effet d'une reprise à chaud (%S1=1) d'un arrêt de l'automate (STOP)	N'a aucun effet sur la valeur courante du compteur (%Ci.V).
Effet de la modification de la valeur de présélection de %Ci.P	La modification de la valeur de présélection à l'aide d'une instruction ou d'un réglage ne prend effet qu'au moment du traitement du bloc par l'application (activation de l'une des entrées).

## Programmation et configuration des compteurs

**Introduction** L'exemple suivant illustre un compteur permettant de compter un maximum de 5000 articles. Chaque impulsion sur l'entrée %I1.2 (lorsque le bit interne %M0 est mis à 1) incrémente la valeur du compteur %C8 d'une unité, jusqu'à la valeur de présélection finale (bit %C8.D=1). Le compteur est remis à zéro par l'entrée %I1.1.

**Exemple de programmation** L'illustration suivante représente un bloc fonction compteur et affiche des exemples de programmation réversible et non réversible.

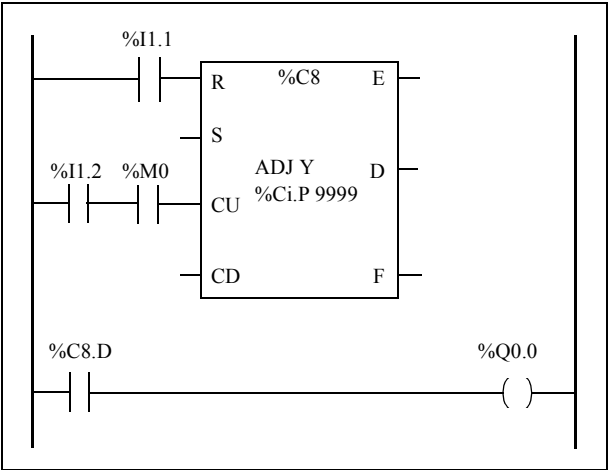


Schéma à contacts

BLK	%C8
LD	%I1.1
R	
LD	%I1.2
AND	%M0
CU	
END_BLK	
LD	%C8.D
ST	%Q0.0

Programmation réversible

LD	%I1.1
R	%C8
LD	%I1.2
AND	%M0
CU	%C8
LD	%C8.D
ST	%Q0.0

Programmation non réversible

**Configuration**

Les paramètres suivants doivent être saisis au moment de la configuration :

- Valeur de présélection (%Ci.P) : fixée à 5000 dans cet exemple
- Réglage : Oui

**Exemple d'un  
compteur/  
Décompteur**

L'illustration suivante représente un bloc fonction compteur / décompteur.

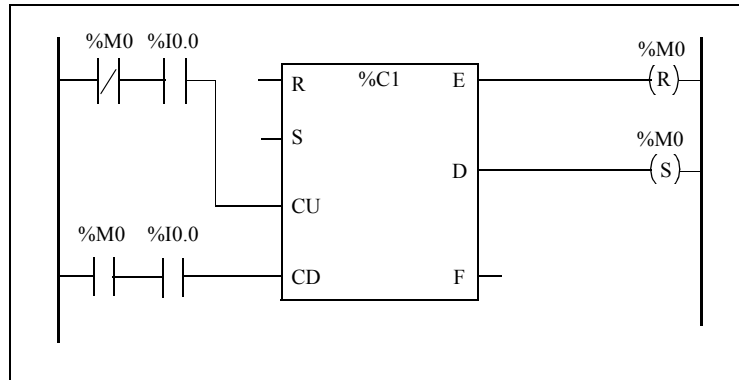


Schéma à contacts

Dans cet exemple, si on prends %C1.P 4, la valeur courante du compteur %C1.V sera incrémenté de 0 jusqu'à 3 puis décrémenté de 3 jusqu'à 0. Tant que %I0.0=1 %C1.V oscille entre 0 et 3.

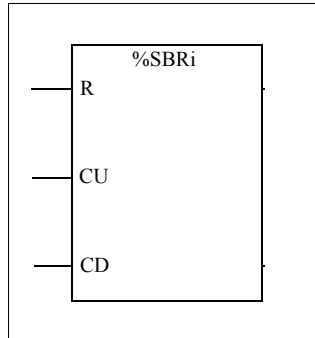
## Bloc fonction registre bits à décalage (%SBRi)

### Introduction

Le bloc fonction registre bits à décalage (%SBRi) effectue un décalage vers la gauche ou vers la droite des bits de données binaires (0 ou 1).

### Illustration

L'exemple suivant illustre un bloc fonction registre à décalage :

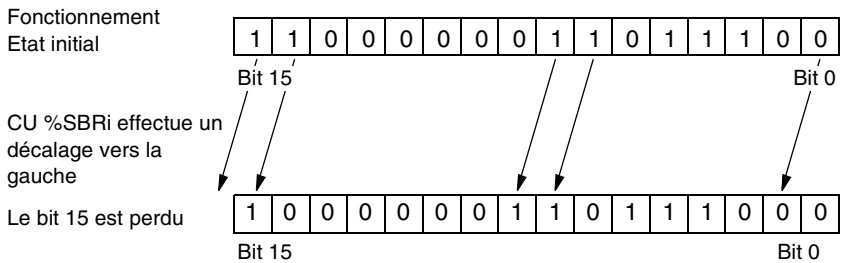


### Paramètres

Le bloc fonction registre bits à décalage possède les paramètres suivants :

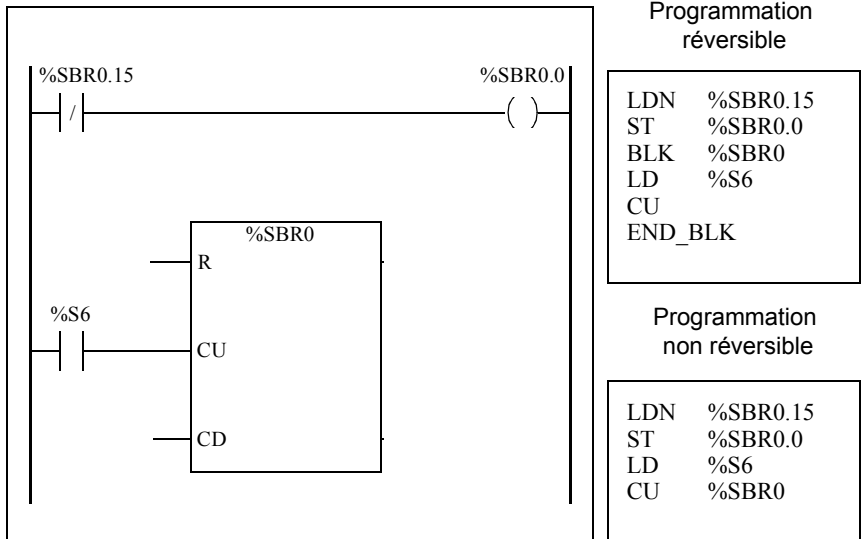
Paramètre	Etiquette	Valeur
Numéro de registre	%SBRi	0 à 7
Bit de registre	%SBRi.j	Les bits 0 à 15 (j = 0 à 15) du registre à décalage peut être testé par une instruction de test et écrit à l'aide d'une instruction d'affectation.
Entrée (ou instruction) de présélection	R	Lorsque le paramètre fonction R est 1, ceci définit les bits de registre 0 à 15 %SBRi.j sur 0.
Entrée (ou l'instruction) décalage à gauche	CU	Sur un front montant, décale un bit du registre vers la gauche.
Entrée (ou l'instruction) décalage à droite	CD	Sur un front montant, décale un bit du registre vers la droite.

**Fonctionnement** L'illustration suivante présente une configuration binaire avant et après une opération de décalage.



Cet exemple peut également s'appliquer à une requête de décalage d'un bit vers la droite (Bit 15 à Bit 0) à l'aide de l'instruction CD. Le bit 0 est perdu.  
Si un registre de 16 bits n'est pas adapté, il est possible d'utiliser le programme pour afficher en cascade plusieurs registres.

**Programmation** Dans l'exemple suivant, un bit est décalé vers la gauche à chaque seconde et le bit 0 prend l'état opposé au bit 15.





**Cas particuliers**

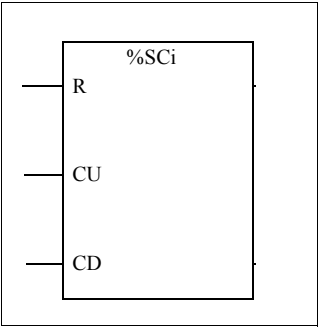
Le tableau suivant présente une liste des cas spéciaux de fonctionnement.

Cas spécial	Description
Effet d'un redémarrage à froid (%S0=1)	Règle tous les bits du mot registre sur 0.
Effet d'une reprise à chaud (%S1=1)	N'a aucun effet sur les bits du mot registre.

## Bloc fonction pas à pas (%SCi)

**Introduction** Un bloc fonction pas à pas (%SCi) permet d'accomplir une série d'étapes auxquelles des actions peuvent être affectées. Le passage d'une étape à l'autre dépend d'événements internes ou externes. Chaque fois qu'une étape est active, le bit associé est réglé sur 1. Une seule étape d'une fonction pas à pas peut être active à la fois.

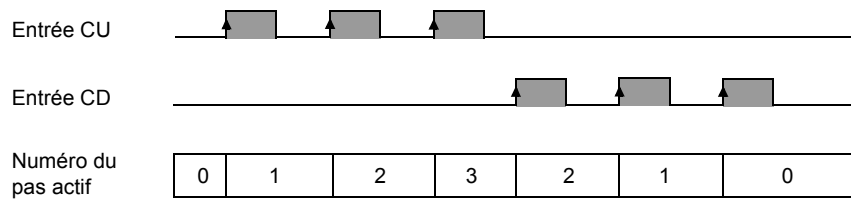
**Illustration** L'exemple suivant illustre un bloc fonction pas à pas.



**Paramètres** Le bloc fonction pas à pas possède les paramètres suivants :

Paramètre	Etiquette	Valeur
Numéro de fonction pas à pas	%SCi	0 - 7
Bit de fonction pas à pas	%SCi.j	Les bits de fonction pas à pas 0 à 255 (j = 0 à 255) peuvent être testés par une instruction logique de chargement et écrits à l'aide d'une instruction d'affectation.
Entrée (ou instruction) de présélection	R	Lorsque le paramètre fonction R est 1, ceci réinitialise la fonction pas à pas.
Entrée (ou instruction) d'incrémentement	CU	Sur un front montant, incrémente la fonction pas à pas d'une étape.
Entrée (ou instruction) de décrémentation	CD	Sur un front montant, décrémente la fonction pas à pas d'une étape.

**Chronogramme** Le chronogramme suivant illustre le fonctionnement du bloc fonction pas à pas.

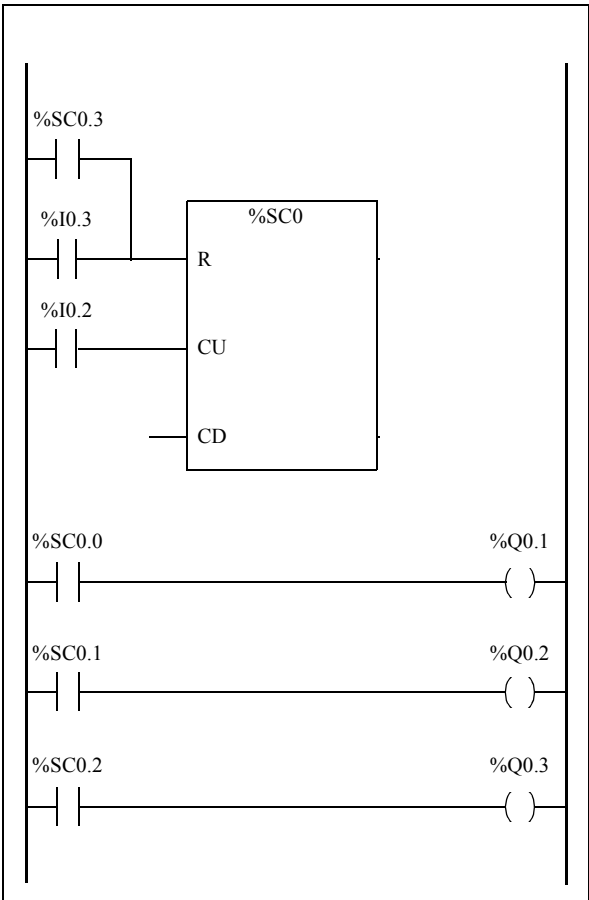


Programmation

L'exemple suivant illustre un bloc fonction pas à pas.

- La fonction pas à pas 0 est incrémentée par l'entrée %I0.2.
- La fonction pas à pas 0 est remise à 0 par l'entrée %I0.3 ou lorsqu'elle arrive à l'étape 3.
- L'étape 0 commande la sortie %Q0.1, l'étape 1 commande la sortie %Q0.2 et l'étape 2 commande la sortie %Q0.3.

L'illustration suivante présente la programmation réversible et non réversible correspondant à cet exemple.



Programmation réversible

```
BLK  %SC0
LD   %SC0.3
OR   %I0.3
R
LD   %I0.2
CU
END _BLK
LD   %SC0.0
ST   %Q0.1
LD   %SC0.1
ST   %Q0.2
LD   %SC0.2
ST   %Q0.3
```

Programmation non réversible

```
LD   %SC0.3
OR   %I0.3
R   %SC0
LD   %I0.2
CU   %SC0
LD   %SC0.0
ST   %Q0.1
LD   %SC0.1
ST   %Q0.2
LD   %SC0.2
ST   %Q0.3
```

**Cas spécifique**

Le tableau suivant présente une liste des cas spécifiques de fonctionnement du bloc fonction pas à pas.

Cas spécifique	Description
Effet d'un redémarrage à froid (%S0=1)	Initialise la fonction pas à pas.
Effet d'une reprise à chaud (%S1=1)	N'a aucun effet sur la fonction pas à pas.

## 14.3            Traitement numérique

---

### Présentation

---

**Objet de ce sous-chapitre**    Ce sous-chapitre offre une introduction au traitement numérique, qui s'appuie sur des descriptions et des directives de programmation.

---

**Contenu de ce sous-chapitre**    Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Introduction aux instructions numériques	359
Instructions d'affectation	360
Instructions de comparaison	365
Instructions arithmétiques sur entiers	367
Instructions logiques	370
Instructions de décalage	372
Instructions de conversion	374
Instructions de conversion entre mots simples et doubles	376

---

## Introduction aux instructions numériques

---

### Présentation

Les instructions numériques s'appliquent généralement aux mots de 16 bits (voir section *Objets mots*, p. 30) et aux doubles mots de 32 bits (Voir *Objets flottants et mots doubles*, p. 34). Ces instructions apparaissent entre crochets. Si le résultat de l'opération logique précédente est Vraie (accumulateur booléen = 1), l'instruction numérique est exécutée. Si ce résultat est Faux (accumulateur booléen = 0), l'instruction numérique n'est pas exécutée et l'opérande reste inchangé.

---

## Instructions d'affectation

---

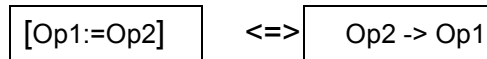
### Introduction

Les instructions d'affectation permettent de charger l'opérande Op2 dans l'opérande Op1.

---

### Affectation

Syntaxe des instructions d'affectation



Les opérations d'affectation peuvent être exécutées sur :

- des chaînes de bits
  - Mots
  - Doubles mots
  - Flottants
  - des tables de mots
  - des tables de doubles mots
  - des tables de flottants
- 

### Affectation de chaînes de bits

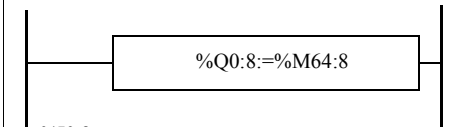
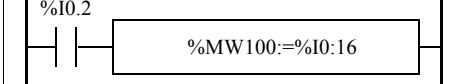
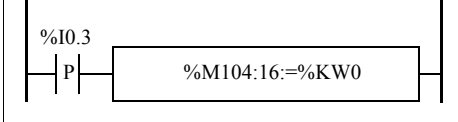
Les opérations peuvent être exécutées sur les chaînes de bits suivantes (voir sous-chapitre *Objets structurés*, p. 48) :

- Chaîne de bit -> chaîne de bit (Exemple 1)
  - Chaîne de bit -> mot (Exemple 2) ou double mot (indexé)
  - Mot ou double mot (indexé) -> chaîne de bit (Exemple 3)
  - Valeur immédiate -> chaîne de bit
-



## Exemples

## Exemples d'affectations de chaînes de bits

	<pre>LD      1 [%Q0:8:=%M64:8]      (Ex. 1)</pre>
	<pre>LD      %I0.2 [%MW100:=%I0:16]      (Ex. 2)</pre>
	<pre>LDR     %I0.3 [%M104:16:=%KW0]      (Ex. 3)</pre>

## Règles d'utilisation :

- Pour l'affectation chaîne de bit -> mot : les bits de la chaîne sont transférés vers le mot en commençant par la droite (premier bit de la chaîne vers bit 0 du mot) et les bits mot non concernés par le transfert (longueur ≤16) sont réglés sur 0.
- Pour l'affectation mot -> chaîne de bits : les bits mot sont transférés en partant de la droite (bit mot 0 vers premier bit de la chaîne).

## Affectations de chaînes de bits

## Syntaxe des affectations de chaînes de bits

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérande 2 (Op2)
:=	[Op1 := Op2 ]  L'opérande 1 (Op1) prend la valeur de l'opérande 2 (Op2).	%MWi,%QWi, %QWai,%SWi %MWi[%MWi], %MDi, %MDi[%MWi] <b>%Mi:L, %Qi:L, %Si:L,            %Xi:L</b>	Valeur immédiate, %MWi, %KWj, %IW,%Iwai, %INWi, %QWi, %QWai %QNWj, %SWi, %BLK.x, %MWi[%MWi], %KWj[%MWi], %MDi[%MWi], %KDi[%MWi], <b>%Mi:L,%Qi:L, %Si:L,            %Xi:L, %li:L</b>

**Note** : L'abréviation %BLK.x (%C0.P, par exemple) est utilisée pour décrire tout mot de bloc fonction.

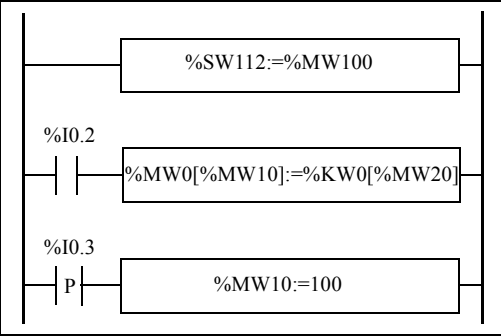
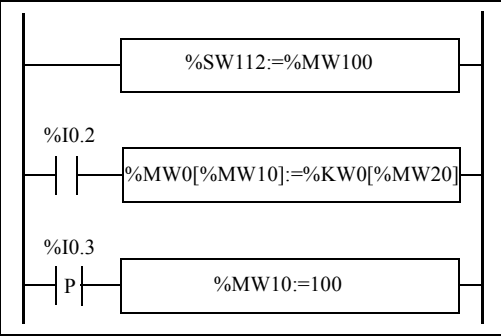
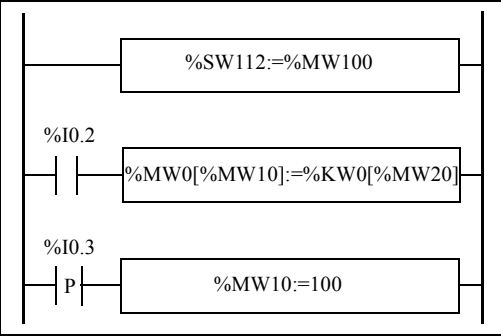
Affectation de mots

Les opérations d'affectation peuvent être exécutées sur les mots et doubles mots suivants :

- Mot (indexé) -> mot (Exemple 2) (indexé ou non)
- Double mot (indexé) -> Double mot (indexé ou non)
- Valeur entière immédiate -> mot (Exemple 3) ou double mot (indexés ou non)
- Chaîne de bit -> mot ou double mot
- Flottant (indexé ou non)-> flottant (indexé ou non)
- Mot ou double mot -> chaîne de bit
- Valeur flottante immédiate -> flottant (indexé ou non)

Exemples

Exemples d'affectations de mots

	<pre>LD      1 [%SW112:=%MW100]    (Ex. 1)</pre>
	<pre>LD      %I0.2 [%MW0[%MW10]:= %KW0[%MW20]]    (Ex. 2)</pre>
	<pre>LDR     %I0.3 [%MW10:=100]    (Ex. 3)</pre>

## Syntaxe

### Syntaxe des affectations de mots

Opérateur	Syntaxe
:=	[Op1 := Op2 ] L'opérande 1 (Op1) prend la valeur de l'opérande 2 (Op2).

Le tableau suivant détaille les opérandes :

Type	Opérande 1 (Op1)	Opérande 2 (Op2)
mot, double mot, chaîne de bits	%BLK.x, %MWi, %QWi, %QWai, %SWi %MWi[MWi, %MDi, %MDi[%MWj]], %Mi:L, %Qi:L, %Si:L, %Xi:L	Valeur immédiate, %MWi, %KWi, %IW, %IWai, %QWi, %QWai, %SWi, %MWi[MWi], %KWi[MWi], %MDi, %MDi[%MWj], %KDi, %KDi[MWj] , %INW, %Mi:L, %Qi:L, %QNW, %Si:L, %Xi:L, %li:L
Flottant	%MFi, %MFi[%MWj]	valeur flottante immédiate, %MFi, %MFi[%MWj], %KFi, %KFi[%MWj]

**Note** : L'abréviation %BLK.x (%R3.I, par exemple) est utilisée pour décrire tout mot de bloc fonction. Pour les chaînes de bits %Mi:L, %Si:L et %Xi:L, le repère de base du premier bit de la chaîne doit être un multiple de 8 (0, 8, 16, ..., 96, ...).

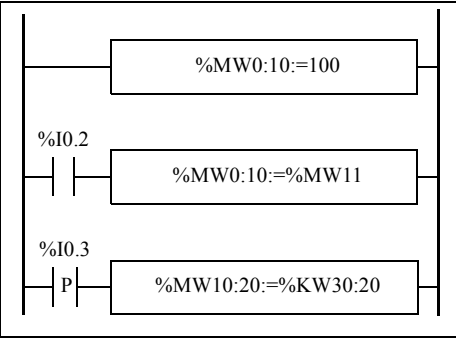
### Affectation de tables de mots, doubles mots ou flottants

Les opérations d'affectation peuvent être exécutées sur les tables d'objets suivantes (voir sous-chapitre *Tables de mots*, p. 49) :

- Valeur entière immédiate -> table de mots (Exemple 1) ou de mots doubles
- Mot -> table de mots (Exemple 2)
- Table de mots -> table de mots (Exemple 3)  
La longueur de la table (L) doit être la même pour les deux tables.
- Double mot -> table de doubles mots
- Table de doubles mots -> table de doubles mots  
La longueur de la table (L) doit être la même pour les deux tables.
- Valeur flottante immédiate -> table de flottants
- Flottant -> table de flottants
- Table de flottants -> table de flottants  
La longueur de la table (L) doit être la même pour les deux tables.

Exemples

Exemples d'affectations de tables de mots



LD 1  
[%MW0:10:=100] (Ex. 1)

LD %I0.2  
[%MW0:10:=%MW11] (Ex. 2)

LDR %I0.3  
[%MW10:20:=%KW30:20] (Ex. 3)

Syntaxe

Syntaxe des affectations de tables de mots, doubles mots et flottants

Opérateur	Syntaxe
:=	[Op1 := Op2 ] L'opérande 1 (Op1) prend la valeur de l'opérande 2 (Op2).

Le tableau suivant détaille les opérandes :

Type	Opérande 1 (Op1)	Opérande 2 (Op2)
tableau de mots	%MWi:L, %SWi:L	%MWi:L, %SWi:L, Valeur entière immédiate, %MWi, %KWi, %IW, %QW, %IWA, %QWA, %SWi, %BLK.x
tableau de doubles mots	%MDi:L	valeur entière immédiate, %MDi, %KDi,%MDi:L, %KDi:L
tableau de flottants	%MFi:L]	valeur flottante immédiate, %MFi, %KFi, %MFi:L, %KFi:L

**Note :** L'abréviation %BLK.x (%R3.I, par exemple) est utilisée pour décrire tout mot de bloc fonction.

## Instructions de comparaison

**Introduction** Les instructions de comparaison permettent de comparer deux opérandes. Le tableau suivant répertorie les différents types d'instructions de comparaison.

Instruction	Fonction
>	Teste si l'opérande 1 est supérieur à l'opérande 2.
>=	Teste si l'opérande 1 est supérieur ou égale à l'opérande 2.
<	Teste si l'opérande 1 est inférieur à l'opérande 2.
<=	Teste si l'opérande 1 est inférieur ou égal à l'opérande 2.
=	Teste si l'opérande 1 est égal à l'opérande 2.
<>	Teste si l'opérande 1 est différent de l'opérande 2.

**Structure** La comparaison s'effectue entre les crochets qui suivent les instructions LD, AND et OR. Le résultat est 1 lorsque le résultat de la comparaison requise est Vrai. Exemples d'instructions de comparaison

```
graph LR
    R1["%MW10>100"] --- Q03["( ) %Q0.3"]
    R2["%M0"] --- B2["%MW20<%KW35"] --- Q02["( ) %Q0.2"]
    R3["%I0.2"] --- B3["%MF30>=%MF40"] --- Q04["( ) %Q0.4"]
```

```
LD  [%MW10 > 100]
ST  %Q0.3

LD  %M0
AND [%MW20 < %KW35]
ST  %Q0.2

LD  %I0.2
OR  [%MF30 >=%MF40]
ST  %Q0.4
```

Syntaxe

Syntaxe des instructions de comparaison :

Opérateur	Syntaxe
>, >=, <, <=, =, <>	LD [Op1 Opérateur Op2] AND [Op1 Opérateur Op2] OR [Op1 Opérateur Op2]

Opérandes :

Type	Opérande 1 (Op1)	Opérande 2 (Op2)
Mots	%MWi, %KWi, %INWi, %IW, %IWAi, %QNWi, %QWi, %QWAI, %QNWi, %SWi, %BLK.x	Valeur immédiate, %MWi, %KWi, %INWi, %IW, %IWAi, %QNWi, %QW, %QWAI, %SWi, %BLK.x, %MWi [%MWi], %KWi [%MWi]
Doubles mots	%MDi, %KDi	Valeur immédiate, %MDi, %KDi, %MDi [%MWi], %KD [%MWi]
Flottants	%MFi, %KFi	Valeur flottante immédiate, %MFi, %KFi, %MFi [%MWi], %KFi [%MWi]

**Note** : Les instructions de comparaison peuvent être utilisées au sein de parenthèses.

Exemple d'utilisation d'une instruction de comparaison entre parenthèses

```
LD      %M0
AND(    [%MF20 > 10.0]
OR      %I0.0
)
ST      %Q0.1
```

## Instructions arithmétiques sur entiers

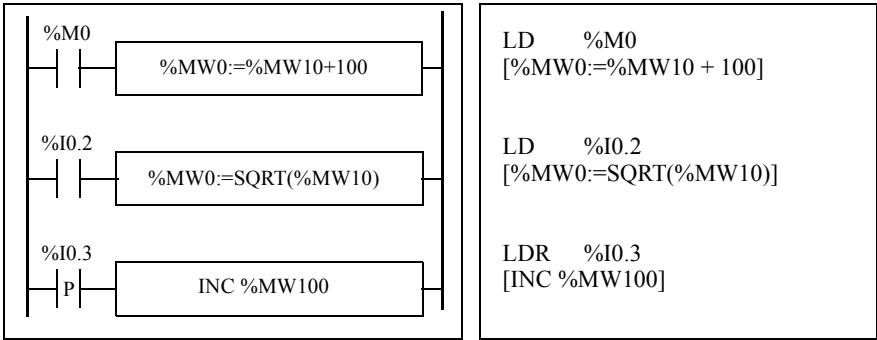
### Introduction

Les instructions arithmétiques permettent d'effectuer des opérations arithmétiques entre deux opérandes entiers ou sur un opérande entier.  
Le tableau suivant répertorie les différents types d'instructions arithmétiques.

Instruction	Fonction
+	Addition de deux opérandes
-	Soustraction de deux opérandes
*	Multiplication de deux opérandes
/	Division de deux opérandes
REM	Reste de la division de deux opérandes
SQRT	Racine carrée d'un opérande
INC	Incrémentation d'un opérande
DEC	Décrémentation d'un opérande
ABS	Valeur absolue d'un opérande

### Structure

Les opérations arithmétiques sont effectuées de la façon suivante :



**Syntaxe**

La syntaxe dépend des opérateurs utilisés, tel que l'indique le tableau ci-dessous.

Opérateur	Syntaxe
+, -, *, /, REM	[Op1: = Op 2 Opérateur Op3]
INC, DEC	[Opérateur Op1]
SQRT (1)	[Op1: = SQRT(Op2)]
ABS (1)	[Op1: = ABS(Op2)]

Opérandes :

Type	Opérande 1 (Op1)	Opérandes 2 et 3 (Op2 & 3) (1)
Mots	%MWi, %QWi, %QWai, %SWi	Valeur immédiate , %MWi, %KW, %INW, %IW, %IWAi, %QNW, %QW, %QWai, %SWi, %BLK.x
Doubles mots	%MDi	Valeur immédiate, %MDi, %KDi

**Note :** (1) Avec cet opérateur, Op2 ne peut pas être une valeur immédiate. La fonction ABS n'est utilisable qu'avec des doubles mots (%MD et %KD) et des flottants (%MF et %KF). Par conséquent OP1 et OP2 doivent être des doubles mots ou des flottants.

**Débordement et conditions d'erreurs****Addition**

- Débordement pendant l'opération sur mots  
Si le résultat dépasse les limites de -32 768 ou de +32 767, le bit %S18 (débordement) est mis à 1. Le résultat est alors non significatif (voir Exemple 1 page suivante). Le programme utilisateur gère le bit %S18.

Remarque :

Pour les doubles mots, les limites sont -2 147 483 648 et 21 474 836 487.

**Multiplication**

- Débordement pendant l'opération  
Si le résultat dépasse la capacité du mot de résultat, le bit %S18 (débordement) est mis à 1 et le résultat n'est pas significatif.

**Division / reste**

- Division par 0  
Si le dividende est 0, la division est impossible et le bit système %S18 est mis à 1. Le résultat est alors incorrect.
- Débordement pendant l'opération



Si le quotient de la division dépasse la capacité du mot de résultat, le bit système %S18 est mis à 1.

### Calcul de la racine carrée

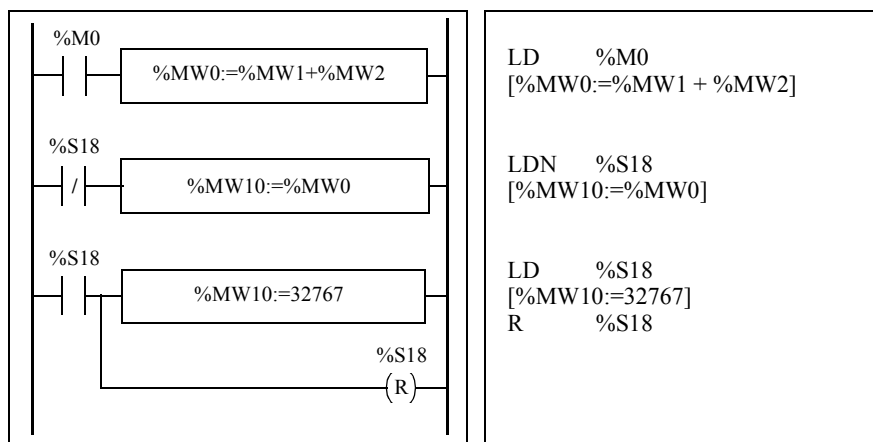
- Débordement pendant l'opération

Le calcul de la racine carrée est uniquement effectué sur les valeurs positives. Le résultat est, par conséquent, toujours positif. Si l'opérande de racine carrée est négatif, le bit système %S18 est mis à 1 et le résultat est incorrect.

**Note :** Le programme utilisateur gère les bits système %S17 et %S18. L'automate les règle sur 1. Ils doivent être remis à 0 par le programme afin de pouvoir être réutilisés (voir exemple page précédente).

## Exemples

### Exemple 1 : débordement lors de l'addition



Si %MW1 =23 241 et %MW2=21 853, le résultat réel (45 094) ne peut pas être exprimé par un mot de 16 bits, le bit %S18 est réglé sur 1 et le résultat obtenu (-20 442) est incorrect. Dans cet exemple, la valeur est fixée à 32 767 lorsque le résultat est supérieur à cette valeur.

## Instructions logiques

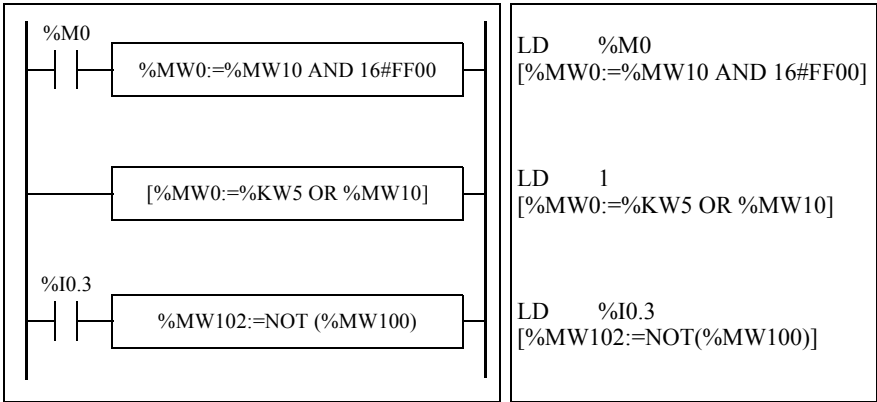
### Introduction

Les instructions logiques permettent d'effectuer des opérations logiques entre deux opérandes ou sur un opérande.  
Le tableau suivant répertorie les différents types d'instructions logiques :

Instruction	Fonction
AND	AND (bit à bit) entre deux opérandes
OR	OR logique (bit à bit) entre deux opérandes
XOR	OR exclusif (bit à bit) entre deux opérandes
NOT	Complément logique (bit à bit) d'un opérande

### Structure

Les opérations logiques sont effectuées de la façon suivante :



**Syntaxe**

La syntaxe dépend des opérateurs utilisés :

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérandes 2 et 3 (Op2 et 3)
AND, OR, XOR	[Op1: = Op2 Opérateur Op3]	%MWi, %QWi, %QWAI, %SWi	Valeur immédiate (1), %MWi, %KWi, %IW, %IWAi, %QW, %QWAI, %SWi, %BLK.x
NOT	[Op1:=NOT(Op2)]		

**Note :** (1) Avec NOT, Op2 ne peut pas être une valeur immédiate.

**Exemple**

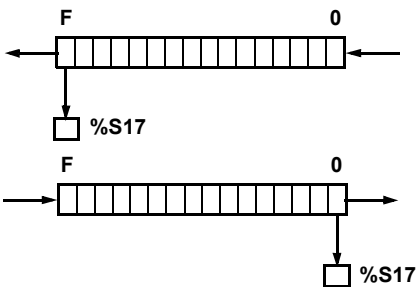
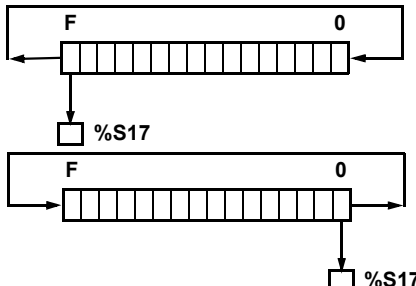
L'exemple suivant présente une instruction AND logique.

```
[%MW15 := %MW32 AND %MW12]
```

## Instructions de décalage

### Introduction

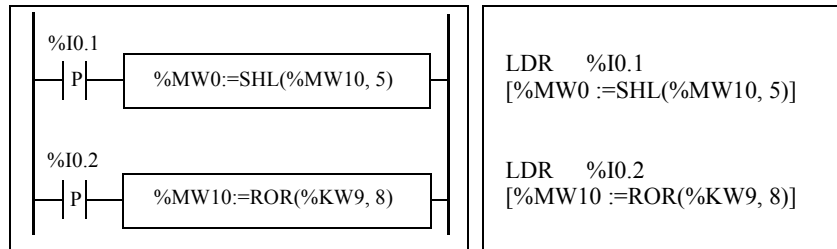
Les instructions de décalage déplacent les bits d'un opérande d'un certain nombre de positions vers la droite ou vers la gauche.  
Le tableau suivant répertorie les différents types d'instructions de décalage.

Instruction	Fonction	
Décalage logique		
SHL(op2,i)	Décalage logique de i positions vers la gauche	
SHR(op2,i)	Décalage logique de i positions vers la droite	
Décalage circulaire		
ROL(op2,i)	Décalage circulaire de i positions vers la gauche	
ROR(op2,i)	Décalage circulaire de i positions vers la droite	

**Note :** Le bit système %S17 (Voir *Bits système (%S)*, p. 546) est utilisé pour le dépassement de capacité.

**Structure**

Les opérations de décalage sont effectuées de la façon suivante :

**Syntaxe**

La syntaxe dépend des opérateurs utilisés, tel que l'indique le tableau ci-dessous.

Opérateur	Syntaxe
SHL, SHR	[Op1 : = Opérateur (Op2,i)]
ROL, ROR	

Opérandes :

Types	Opérande 1 (Op1)	Opérande 2 (Op2)
Mots	%MWi, %QWi, %QWai, %SWi	%MWi, %KWi, %IW, %Iwai, %QW, %QWai, %SWi, %BLK.x
Double mots	%MDi	%MDi, %KDi

## Instructions de conversion

### Introduction

Les instructions de conversion permettent d'effectuer la conversion entre les différentes représentations numériques.  
Le tableau suivant répertorie les différents types d'instructions de conversion.

Instruction	Fonction
BTI	BCD --> Conversion binaire
ITB	Binaire --> Conversion BCD

### Révision du code BCD

Le codage BCD (Binary Coded Decimal - décimal codé binaire) représente les décimaux (entre 0 et 9) par un code à quatre bits. Un objet mot de 16 bits peut ainsi contenir un nombre exprimé par quatre chiffres (0000 - 9999), et un objet double mot de 32 bits peut ainsi contenir un nombre exprimé par huit chiffres.  
Lors d'une conversion, le bit système %S18 est mis à 1 si la valeur n'est pas BCD. Ce bit doit être testé et remis à 0 par le programme.  
Représentation BCD des décimaux :

Décimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

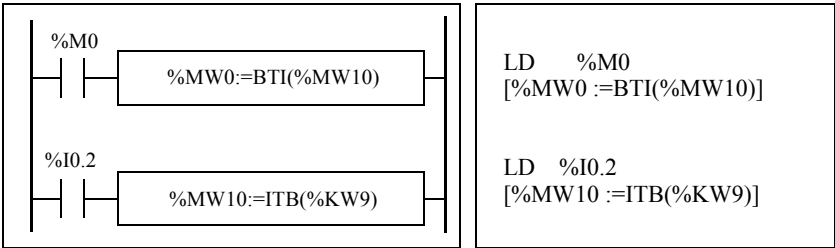
Exemples :

- Le mot %MW5 exprime la valeur BCD "2450", qui correspond à la valeur binaire : 0010 0100 0101 0000
- Le mot %MW12 exprime la valeur décimale "2450", qui correspond à la valeur binaire : 0000 1001 1001 0010

Le mot %MW5 est converti en mot %MW12 à l'aide de l'instruction BTI.  
Le mot %MW12 est converti en mot %MW5 à l'aide de l'instruction ITB.

### Structure

Les opérations de conversion sont effectuées de la façon suivante :



**Syntaxe**

La syntaxe dépend des opérateurs utilisés, tel que l'indique le tableau ci-dessous.

Opérateur	Syntaxe
BTI, ITB	[Op1 := Opérateur (Op2)]

Opérandes :

Type	Opérande 1 (Op1)	Opérande 2 (Op2)
Mots	%MWi, %QWi, %QWai, %SWi	%MWi, %KWi, %IW, %IWAi, %QW, %QWai, %SWi, %BLK.x
Mots double	%MDi	%MDi, %KDi

**Exemples  
d'application :**

L'instruction BTI peut être utilisée pour traiter une valeur de consigne aux entrées de l'automate via des roues codeuses en BCD.

L'instruction peut être utilisée pour afficher des valeurs numériques sur des afficheurs codés en BCD (résultat d'un calcul, valeur courante d'un bloc fonction, par exemple).

Instructions de conversion entre mots simples et doubles

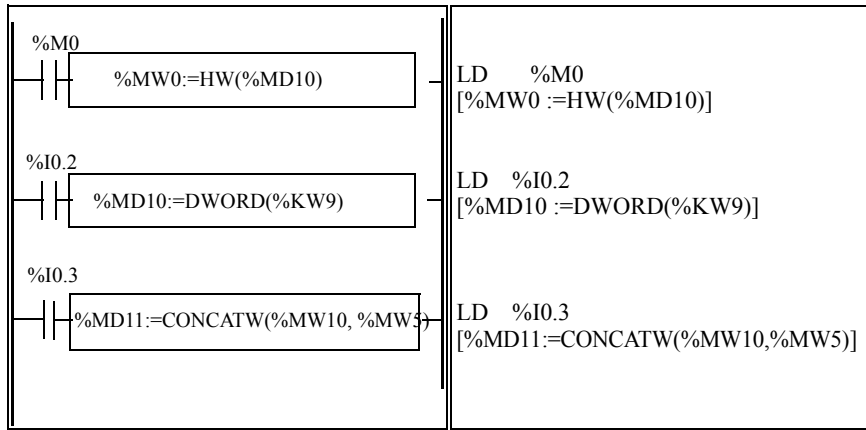
Inroduction

Le tableau suivant décrit les instructions de conversions entre les mots simples et doubles :

Instruction	Fonction
LW	Extrait l'octet de poids faible d'un double mot vers un mot.
HW	Extrait l'octet de poids fort d'un double mot vers un mot.
CONCATW	Concatène deux mots pour constituer un double mot.
DWORD	Convertit un mot de 16 bits en un double mot de 32 bits.

Structure

Les opérations de conversion sont effectuées de la façon suivante :



Syntaxe

La syntaxe dépend des opérateurs utilisés, tele que l'indique le tableau suivant : l

Opérateur	Syntaxe	Opérande 1 (Op1)	Opérande 2 (Op2)	Opérande 3 (Op3)
LW, HW	Op1 = Opérateur (Op2)	%MWi	%MDi, %KDi	[-]
CONCATW	Op1 = Opérateur (Op2, Op3))	%MDi	%MWi, %KW <sub>i</sub> , valeur immédiate	%MW <sub>i</sub> , %KW <sub>i</sub> , valeur immédiate
DWORD	Op1 = Opérateur (Op2)	%MDi	%MW <sub>i</sub> , %KW <sub>i</sub>	[-]



---

## 14.4 Instructions sur programme

---

### Présentation

---

**Objet de ce sous-chapitre** Ce sous-chapitre présente une introduction aux instructions sur programme.

---

**Contenu de ce sous-chapitre** Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Instructions END	378
Instruction NOP	380
Instructions de saut	381
Instructions de sous-programme	382

---

## Instructions END

---

### Introduction

Les instructions END définissent la fin de l'exécution de la scrutation d'un programme.

---

### END, ENDC et ENDCN

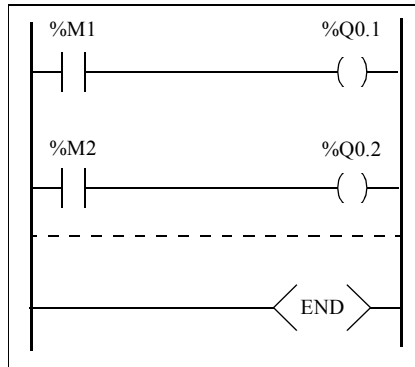
Il existe trois instructions END différentes :

- END : fin de programme inconditionnelle
- ENDC : fin de programme si le résultat booléen de l'instruction sur test précédente est 1
- ENDCN : fin de programme si le résultat booléen de l'instruction sur test précédente est 0

Par défaut (en mode Normal), des sorties sont générées et la scrutation suivante est lancée dès la fin d'un programme.

Si la scrutation est périodique, des sorties sont générées et la scrutation suivante est lancée dès que la fin de période est atteinte.

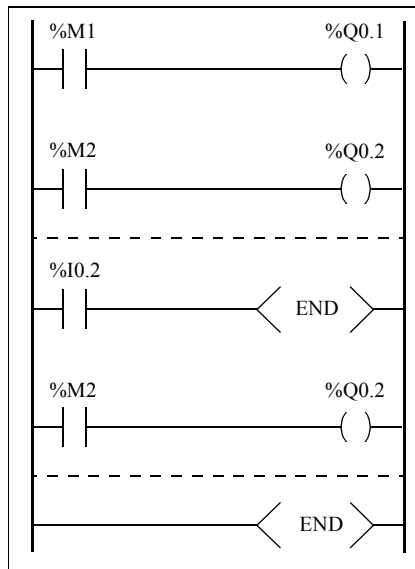
---

**Exemples****Exemple d'instruction END inconditionnelle**

```
LD    %M1
ST    %Q0.1
LD    %M2
ST    %Q0.2
```

.....

END

**Exemple d'instruction END conditionnelle**

```
LD    %M1
ST    %Q0.1
LD    %M2
ST    %Q0.2
```

.....

```
LD    %I0.2
ENDC  → Si %I0.2 = 1, fin de
LD    %M2      scrutation du
ST    %Q0.2    programme
```

Si %I0.2 = 0, continue  
la scrutation du  
programme jusqu'à la  
nouvelle instruction

.....  
END

## Instruction NOP

---

### NOP

L'instruction NOP n'effectue aucune opération. Utilisez cette instruction pour « réserver » des lignes d'un programme afin de pouvoir insérer ultérieurement des instructions, sans modifier les numéros de ligne.

---

## Instructions de saut

### Introduction

Les instructions de saut ont pour effet d'interrompre immédiatement l'exécution d'un programme et de le reprendre à partir de la ligne suivant la ligne contenant l'étiquette %Li (i = 1 à 16 pour un compact et de 1 à 63 pour les autres).

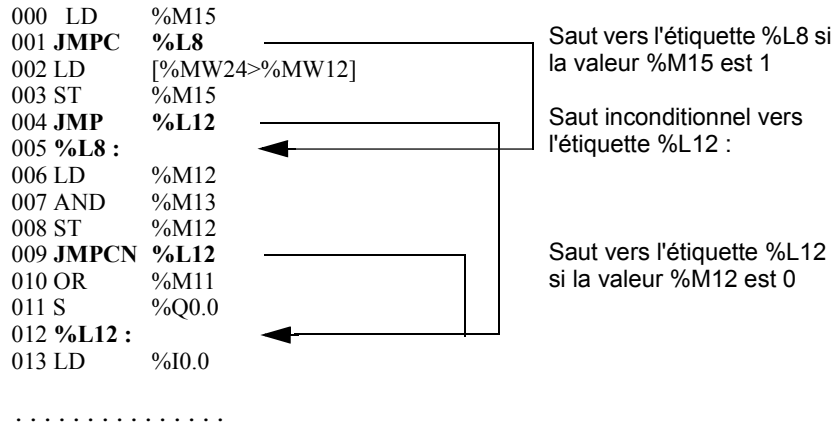
### JMP, JMPC et JMPCN

Trois instructions de saut différentes sont disponibles :

- **JMP** : saut de programme inconditionnel
- **JMPC** : saut de programme si le résultat booléen de la logique précédente est 1.
- **JMPCN** : saut de programme si le résultat booléen de la logique précédente est 0.

### Exemples

Exemples d'instructions de saut



### Directives

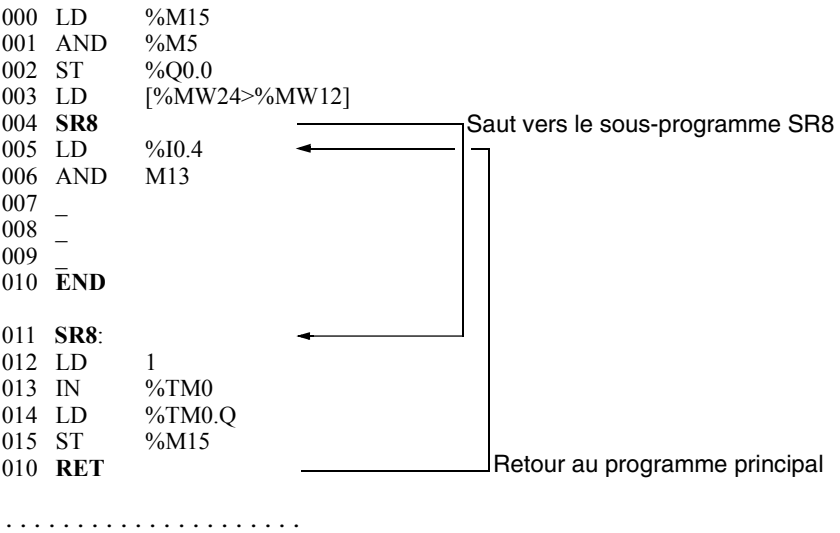
- Les instructions de saut sont interdites entre parenthèses et ne doivent pas être placées entre les instructions AND(, OR( et une parenthèse fermante ")".
- L'étiquette peut uniquement être placée devant une instruction LD, LDN, LDR, LDF ou BLK.
- Le numéro de l'étiquette %Li doit être défini une seule fois dans un programme.
- Le saut de programme est effectué vers une ligne de programmation en amont ou en aval. Lorsque le saut est en amont, le temps de scrutation doit être contrôlé. Un temps de scrutation trop long peut provoquer le déclenchement du chien de garde.

# Instructions de sous-programme

**Introduction** Les instructions de sous-programme déclenchent l'exécution d'un sous-programme, puis le retour vers le programme principal.

- SRn, SRn: et RET** Les sous-programmes se composent de trois étapes :
- L'instruction **SRn** appelle le sous-programme référencé par l'étiquette SRn, si le résultat de l'instruction booléenne précédente est 1.
  - Le sous-programme est référencé par l'étiquette **SRn:**, n pouvant prendre une valeur comprise entre 0 à 15 pour TWDLCAA10DRF, TWDLCAA16DRF et 0 à 63 pour tous les autres automates.
  - L'instruction **RET** placée à la fin du sous-programme provoque le retour au programme principal.

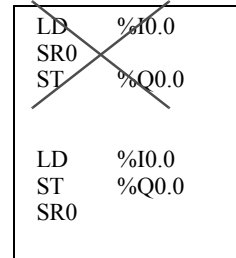
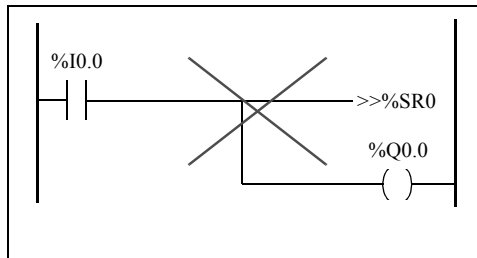
**Exemple** Exemples d'instructions de sous-programme



**Directives**

- Un sous-programme ne doit pas appeler un autre sous-programme.
- Les instructions de sous-programme sont interdites entre parenthèses et ne doivent pas être placées entre les instructions AND(, OR( et une fermeture de parenthèse ")".
- L'étiquette peut uniquement être placée devant une instruction LD ou BLK pour marquer le début d'une équation booléenne (ou d'un réseau booléen).
- L'appel du sous-programme ne doit pas être suivi d'une instruction d'affectation. En effet, le sous-programme risque de modifier le contenu de l'accumulateur booléen. Aussi celui risque d'avoir une valeur de retour différente de celle qu'il avait avant l'appel. Voir l'exemple suivant.

Exemple de programmation d'un sous-programme







---

## Présentation

### Objet de ce chapitre

Ce chapitre offre des informations sur les instructions et les blocs fonctions avancées utilisés pour créer des programmes destinés aux automates Twido.

### Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
15.1	Blocs fonctions avancés	387
15.2	Fonctions horodateur	436
15.3	Fonction PID	448
15.4	Instructions sur flottants	512
15.5	Instructions sur tableaux d'objets	526



## 15.1 Blocs fonctions avancés

### Présentation

**Objet de ce sous-chapitre** Ce sous-chapitre offre une présentation des blocs fonctions avancés et contient des exemples de programmation.

**Contenu de ce sous-chapitre** Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Objets mots et objets bits associés à des blocs fonction avancés	388
Principes de programmation de blocs fonctions avancés	390
Bloc fonction registre LIFO/FIFO (%Ri)	393
LIFO, fonctionnement	395
FIFO, fonctionnement	396
Programmation et configuration des registres	397
Bloc fonction %PWM (modulation de la largeur d'impulsion)	400
Bloc fonction sortie du générateur d'impulsions (%PLS)	404
Bloc fonction programmeur cyclique (%DR)	408
Fonctionnement du bloc fonction programmeur cyclique %DRi	410
Programmation et configuration des programmeurs cycliques	412
Bloc fonction compteur rapide (%FC)	414
Bloc fonction compteur rapide (%VFC)	418
Emission/réception de messages - Instruction d'échange (EXCH)	431
Bloc fonction de contrôle d'échange (%MSGx)	432

## Objets mots et objets bits associés à des blocs fonction avancés

### Introduction

Les blocs fonction avancés utilisent des mots et des bits dédiés de même type que les blocs fonction standards. Les blocs fonction avancés comprennent :

- les registres LIFO/FIFO (%R) ;
- les programmeurs cycliques (%DR) ;
- les compteurs rapides (%FC) ;
- les compteurs rapides (%VFC) ;
- la sortie de modulation de la largeur de l'impulsion (%PWM) ;
- la sortie du générateur d'impulsions (%PLS) ;
- le registre bits à décalage (%SBR) ;
- la fonction pas à pas (%SC) ;
- le bloc contrôle message (%MSG).

### Objets accessibles par le programme

Le tableau suivant présente les mots et les bits accessibles par le programme associés aux différents blocs fonction avancés. Veuillez noter que l'accès en écriture mentionné dans le tableau suivant dépend du paramètre " Réglable ", sélectionné au moment de la configuration. Ce réglage permet d'autoriser ou de refuser l'accès aux mots ou aux bits par TwidoSoft ou par l'interface opérateur.

Bloc fonction avancé	Mots et bits associés		Repère	Accès en mode écriture
%R	Mot	Entrée du registre	%Ri.I	Oui
	Mot	Sortie du registre	%Ri.O	Oui
	Bit	Sortie registre plein	%Ri.F	Non
	Bit	Sortie registre vide	%Ri.E	Non
%DR	Mot	Numéro du pas courant	%DRi.S	Oui
	Bit	Dernier pas égal au pas courant	%DRi.F	Non
%FC	Mot	Valeur courante	%FCi.V	Oui
	Mot	Valeur de présélection	%FCi.P	Oui
	Bit	Terminé	%FCi.D	Non

Bloc fonction avancé	Mots et bits associés		Repère	Accès en mode écriture
%VFC	Mot	Valeur courante	%VFCi.V	Non
	Mot	Valeur de présélection	%VFCi.P	Oui
	Bit	Sens de comptage	%VFCi.U	Non
	Mot	Valeur de capture	%VFCi.C	Non
	Mot	Valeur de seuil 0	%VFCi.S0	Oui
	Mot	Valeur de seuil 1	%VFCi.S1	Oui
	Bit	Sortie pour	%VFCi.F	Non
	Bit	Sortie réflexe 0 activée	%VFCi.R	Oui
	Bit	Sortie réflexe 1 activée	%VFCi.S	Oui
	Bit	Sortie seuil 0	%VFCi.TH0	Non
	Bit	Sortie seuil 1	%VFCi.TH1	Non
	Bit	Base temps de la mesure de fréquence	%VFCi.T	Oui
%PWM	Mot	Pourcentage d'impulsions au pas 1 par rapport à la période totale.	%PWMi.R	Oui
	Mot	Période pré réglée	%PWMi.P	Oui
%PLS	Mot	Nombre d'impulsions	%PLSi.N	Oui
	Mot	Valeur de présélection	%PLSi.P	Oui
	Bit	Sortie courante activée	%PLSi.Q	Non
	Bit	Génération terminée	%PLSi.D	Non
%SBR	Bit	Bit de registre	%SBRi.J	Non
%SC	Bit	Bit de compteur à pas	%SCi.j	Oui
%MSG	Bit	Terminé	%MSGi.D	Non
	Bit	Erreur	%MSGi.E	Non

## Principes de programmation de blocs fonctions avancés

---

### Présentation

Les applications Twido sont stockées sous la forme de programmes par listes, et ce, même si ces applications ont été rédigées à l'aide d'un éditeur schéma à contacts. Les automates Twido peuvent ainsi être considérées comme des "machines à listes". Le terme "réversibilité" se rapporte à la capacité de TwidoSoft à convertir une application liste d'instructions en application schémas à contacts, et vice versa. Par défaut, tous les programmes schémas à contacts sont réversibles.

Tout comme les blocs fonctions élémentaires, les blocs fonctions avancés doivent se conformer à des règles de réversibilité. La structure des blocs fonctions réversibles dans le langage liste d'instructions requiert l'utilisation des instructions suivantes :

- **BLK** : marque le début du bloc et la section d'entrée du bloc fonction.
- **OUT\_BLK** : marque le début de la section de sortie du bloc fonction.
- **END\_BLK** : marque la fin du bloc fonction.

**Note** : Il n'est pas nécessaire d'utiliser ces instructions de blocs fonctions réversibles pour un programme par listes d'instructions qui fonctionne correctement. Certaines instructions permettent une programmation en langage liste d'instructions non réversible.

## Entrées et sorties dédiées

Les fonctions avancées Compteur rapide (FC), Compteur très rapide (VFC), PLS et PWM utilisent des entrées et des sorties dédiées. Ces bits ne sont toutefois pas réservés à une utilisation exclusive par un bloc unique. Il faut donc gérer correctement l'utilisation de ces bits.

Lorsque vous utilisez des fonctions avancées, il est nécessaire que vous gériez la méthode d'allocation des entrées et des sorties dédiées. TwidoSoft vous assiste lors de la configuration de ces ressources en affichant des informations de configuration d'E/S et en vous avertissant si une entrée ou une sortie dédiée est déjà utilisée par un bloc fonction configuré.

Le tableau suivant résume les dépendances des entrées et des sorties dédiées, ainsi que les fonctions spécifiques.

En cas d'utilisation avec des fonctions de comptage :

Entrées	Utilisation
%I0.0.0	%VFC0 : Gestion Haut/Bas ou Phase B
%I0.0.1	%VFC0 : Entrée d'impulsion ou phase A
%I0.0.2	%FC0 : Entrée d'impulsion ou entrée de présélection %VFC0
%I0.0.3	%FC1 : Entrée d'impulsion ou entrée de capture %VFC0
%I0.0.4	%FC2 : Entrée d'impulsion ou entrée de capture %VFC1
%I0.0.5	Entrée de présélection %VFC1
%I0.0.6	%VFC1 : Gestion Haut/Bas ou Phase B
%I0.0.7	%VFC1 : Entrée d'impulsion ou phase A

En cas d'utilisation avec des fonctions de comptage ou des fonctions spéciales :

Sorties	Utilisation
%Q0.0.0	Sortie %PLS0 ou PWM0
%Q0.0.1	Sortie %PLS1 ou PWM1
%Q0.0.2	Sorties réflexes pour %VFC0
%Q0.0.3	
%Q0.0.4	Sorties réflexes pour %VFC1
%Q0.0.5	

### Utilisation d'entrées et de sorties dédiées

TwidoSoft utilise les règles suivantes lors de l'utilisation d'entrées et de sorties dédiées.

- Chaque bloc fonction utilisant des E/S dédiées doit être configuré puis utilisé dans l'application. L'E/S est uniquement allouée lors de la configuration d'un bloc fonction. Elle ne l'est pas lors de son utilisation dans un programme.
- Après qu'un bloc fonction a été configuré, son entrée et sa sortie dédiées ne peuvent pas être utilisées par l'application ou par un autre bloc fonction.  
Par exemple, si vous configurez %PLS0, vous ne pouvez pas utiliser %Q0.0.0 dans %DR0 (programmeur cyclique) ou dans la logique de l'application (ST %Q0.0.0).
- Si une entrée ou une sortie dédiée est requise par un bloc fonction déjà utilisé par l'application ou par un autre bloc fonction, il n'est pas possible de configurer ce bloc fonction.  
Par exemple, si vous configurez %FC0 comme compteur, %VFC0 ne pourra pas être configuré pour utiliser %I0.0.2 comme entrée de capture.

<p><b>Note :</b> Pour modifier l'utilisation des E/S dédiées, vous devez d'abord supprimer la configuration du bloc fonction en définissant le type d'objet sur "non utilisé", puis supprimer les références au bloc fonction dans votre application.</p>
---



## Bloc fonction registre LIFO/FIFO (%Ri)

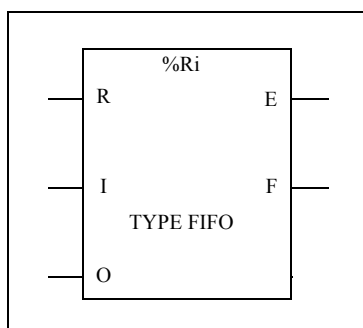
### Introduction

Un registre est un bloc mémoire qui permet de stocker jusqu'à 16 mots de 16 bits de deux manières différentes :

- par une file d'attente, appelée "FIFO" (First In, First Out – Premier entré, Premier sorti) ;
- par une pile, appelée "LIFO" (Last In, First Out – Dernier entré, Premier sorti).

### Illustration

L'exemple suivant illustre l'utilisation du bloc fonction registre.



Bloc fonction registre

**Paramètres**

Le bloc fonction registre possède les paramètres suivants :

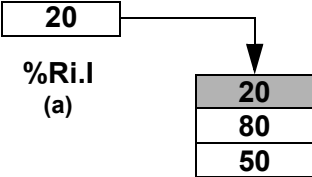
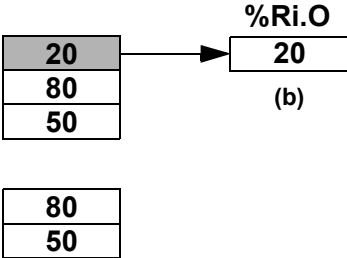
Paramètre	Etiquette	Valeur
Numéro de registre	%Ri	0 à 3
Type	FIFO ou LIFO	File d'attente ou Pile)
Mot d'entrée	%Ri.I	Mot d'entrée du registre. Peut être lu, testé et écrit.
Mot de sortie	%Ri.O	Mot de sortie du registre. Peut être lu, testé et écrit.
Entrée (ou instruction) de stockage	I (In, Entrée)	Sur un front montant, stocke le contenu du mot %Ri.I dans le registre.
Entrée (ou instruction) de récupération	O (Out, Sortie)	Sur un front montant, charge un mot de données du registre dans le mot %Ri.O.
Entrée (ou instruction) RAZ	R (Remise à zéro)	A l'état 1, initialise le registre.
Sortie "Vide"	E (vide)	Le bit %Ri.E associé indique que le registre est vide. Peut être testé.
Sortie "Plein"	F (plein)	Le bit %Ri.F associé indique que le registre est plein. Peut être testé.

---

## LIFO, fonctionnement

**Introduction** En fonctionnement LIFO (Last In, First Out - Dernier entré, Premier sorti), la dernière information entrée est la première à être récupérée.

**Fonctionnement** Le tableau suivant décrit le fonctionnement LIFO.

Etape	Description	Exemple
1	A la réception d'une demande de stockage (front montant sur l'entrée I ou activation de l'instruction I), le contenu du mot d'entrée %Ri.I (qui a préalablement été chargé) est stocké au plus haut de la pile (fig. a). Lorsque la pile est pleine (sortie F=1), plus aucun élément ne peut être stocké.	<p>Stockage du contenu de %Ri.I en haut de la pile.</p> 
2	A la réception d'une demande de récupération (front montant sur l'entrée O ou activation de l'instruction O), le mot de données le plus haut (le dernier à avoir été entré) est chargé dans le mot %Ri.O (fig. b). Lorsque le registre est vide (sortie E=1), plus aucun élément ne peut être récupéré. Le mot de sortie %Ri.O n'est pas modifié et sa valeur reste inchangée.	<p>Récupération du mot de données au plus haut de la pile.</p> 
3	La pile peut être réinitialisée à tout moment (état 1 sur l'entrée R ou activation de l'instruction R). L'élément indiqué par le pointeur est alors le plus haut dans la pile.	

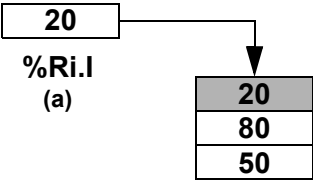
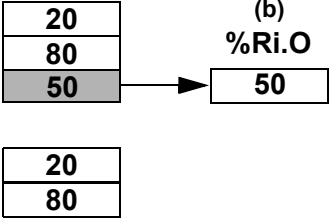
## FIFO, fonctionnement

### Introduction

En fonctionnement FIFO (First In, First Out - Premier entré, Premier sorti), la première information entrée est la première à être récupérée.

### Fonctionnement

Le tableau suivant décrit le fonctionnement FIFO.

Etape	Description	Exemple
1	A la réception d'une demande de stockage (front montant sur l'entrée I ou activation de l'instruction I), le contenu du mot d'entrée %Ri.I (qui a préalablement été chargé) est stocké au plus haut de la file d'attente (fig. a). Lorsque la file d'attente est pleine (sortie F=1), plus aucun élément ne peut être stocké.	Stockage du contenu de %Ri.I en haut de la file d'attente. 
2	A la réception d'une demande de récupération (front montant sur l'entrée O ou activation de l'instruction O), le mot de données le moins haut dans la file d'attente est chargé dans le mot de sortie %Ri.O et le contenu du registre est déplacé d'une place vers le bas, dans la file d'attente (fig. b). Lorsque le registre est vide (sortie E=1), plus aucun élément ne peut être récupéré. Le mot de sortie %Ri.O n'est pas modifié et sa valeur reste inchangée.	Récupération de la première information qui est ensuite chargée dans %Ri.O. 
3	La file d'attente peut être réinitialisée à tout moment (état 1 sur l'entrée R ou activation de l'instruction R).	

## Programmation et configuration des registres

---

### Introduction

L'exemple de programmation suivant illustre le chargement du contenu d'un mot mémoire (%MW34) dans un registre (%R2.I) lors d'une demande de stockage (%I0.2), si le registre %R2 n'est pas plein (%R2.F = 0). La demande de stockage dans le registre est effectuée par %M1. La demande de récupération est effectuée par l'entrée %I0.3 et %R2.O est chargé dans %MW20, si le registre n'est pas vide (%R2.E = 0).

---

Exemple de programmation

L'illustration suivante représente un bloc fonction registre et présente des exemples de programmation réversible et non réversible.

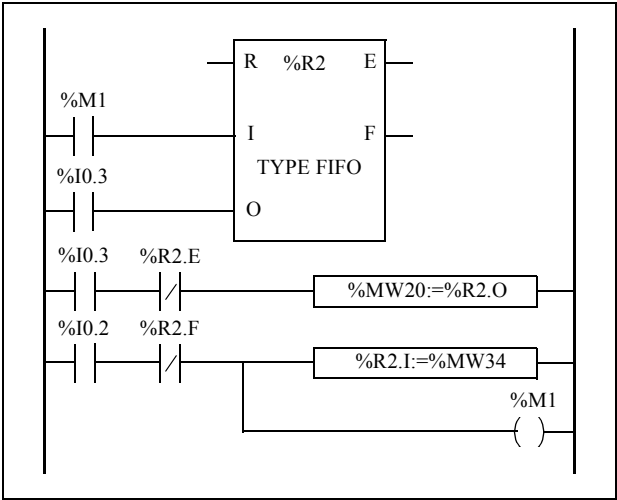


Schéma à contacts

```
BLK      %R2
LD       %M1
I
LD       %I0.3
O
END_BLK
LD       %I0.3
ANDN     %R2.E
[%MW20:=%R2.O]
LD       %I0.2
ANDN     %R2.F
[%R2.I:=%MW34]
ST       %M1
```

Programme réversible

```
LD       %M1
I        %R2
LD       %I0.3
O        %R2
ANDN     %R2.E
[%MW20:=%R2.O]
LD       %I0.2
ANDN     %R2.F
[%R2.I:=%MW34]
ST       %M1
```

Programme non réversible

- Configuration** Seul le type du registre devra être entré au cours de la configuration.
- FIFO (par défaut), ou
  - LIFO

**Cas spécifiques** Le tableau suivant présente une liste de cas spécifiques de fonctionnement :

Cas spécifique	Description
Effet d'un redémarrage à froid (%S0=1)	Provoque l'initialisation du contenu du registre. Le bit de sortie %Ri.E associé à la sortie E est mis à 1.
Effet d'une reprise à chaud (%S1=1) d'un arrêt de l'automate	N'a aucun effet sur la valeur courante du registre ou sur l'état de ses bits de sortie.

## Bloc fonction %PWM (modulation de la largeur d'impulsion)

---

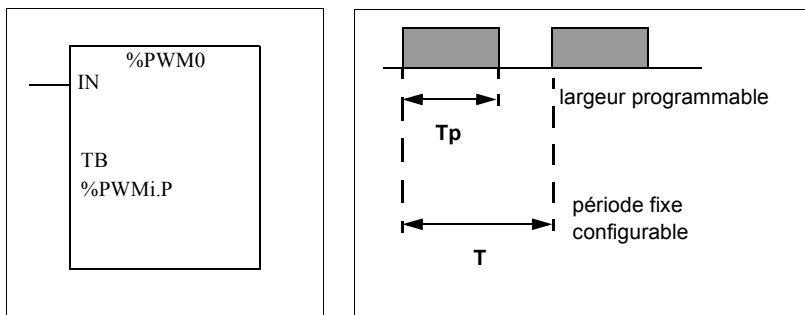
### Introduction

Le bloc fonction de modulation de la largeur d'impulsion (%PWM) génère un signal rectangulaire sur des voies de sortie dédiées (%Q0.0.0 ou %Q0.0.1), dont on peut faire varier la largeur, et, par conséquent, le rapport cyclique. Les automates disposant de sorties relais pour ces deux voies ne prennent pas en charge cette fonction, en raison d'une limitation de fréquences.

Deux blocs %PWM sont disponibles. Le bloc %PWM0 utilise la sortie dédiée %Q0.0.0 et le bloc %PMW1 utilise la sortie dédiée %Q0.0.1. Les blocs fonction %PLS se partagent les mêmes sorties dédiées. Il est donc nécessaire de choisir l'une ou l'autre des fonctions.

### Illustration

Bloc PWM et chronogramme :





**Paramètres**

Le tableau suivant présente les différents paramètres du bloc fonction PWM.

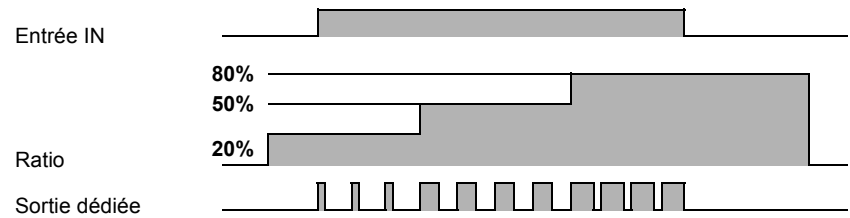
Paramètre	Etiquette	Description
Base temps	TB	0,142 ms, 0,57 ms, 10 ms, 1 s (valeur par défaut)
Présélection de la période	%PWMi.P	<p><math>0 &lt; \%PWMi.P \leq 32767</math> avec une base temps de 10 ms ou 1 s</p> <p><math>0 &lt; \%PWMi.P \leq 255</math> avec une base temps de 0,57 ms ou 0.142 ms</p> <p>0 = Fonction non utilisée</p>
Rapport cyclique	%PWMi.R	<p>Cette valeur donne le pourcentage du signal à l'état 1 au cours d'une période. Le <math>T_p</math> de largeur est ainsi égal à : <math>T_p = T * (\%PWMi.R/100)</math>. L'application utilisateur écrit la valeur de %PWMi.R. Ce mot contrôle le rapport cyclique de la période. Pour plus d'informations sur la définition T, reportez-vous à la section suivante, intitulée "Plage de périodes".</p> <p>La valeur par défaut est 0 et les valeurs supérieures à 100 sont considérées comme étant égales à 100.</p>
Entrée génération de l'impulsion	IN	A l'état 1, le signal de modulation de la largeur d'impulsion est généré sur la voie de sortie. A l'état 0, la voie de sortie est mis à 0.

**Plage de périodes**

La valeur de présélection et la base temps peuvent être modifiées au moment de la configuration. Ces paramètres sont utilisés pour fixer la période du signal  $T = \%PWMi.P * TB$ . L'obtention de rapports bas nécessite que le %PWMi.P sélectionné soit d'autant plus élevé. Plage de périodes disponibles :

- 0,142 ms à 36,5 ms en pas de 0,142 ms (27,4 Hz à 7 kHz)
- 0,57 ms à 146 ms en pas de 0,57 ms (6,84 Hz à 1,75 kHz)
- 10 ms à 5,45 min en pas de 10 ms
- 1 s à 9,1 heures en pas de 1 s

**Fonctionnement** La fréquence du signal de sortie est réglée au moment de la configuration en sélectionnant la base temps et le %PWMi.P pré-réglé. La modification du rapport cyclique % PWMi.R dans le programme permet de moduler la largeur du signal. L'illustration suivante représente un diagramme d'impulsion du bloc fonction PWM avec différents rapports cyclique.



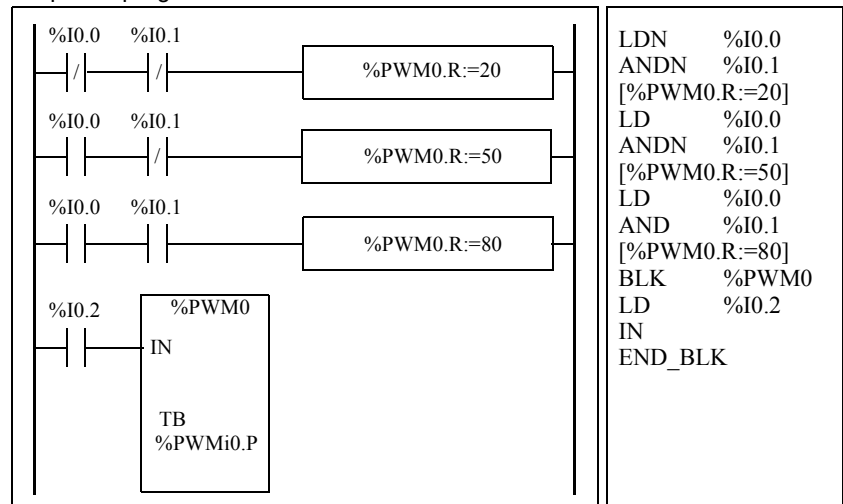
**Programmation et configuration** Dans cet exemple, la largeur du signal est modifiée par le programme en fonction de l'état des entrées %I0.0.0 et %I0.0.1 de l'automate.

Si %I0.0.1 et %I0.0.2 sont réglés sur 0, le rapport %PWM0.R est réglé sur 20 % et la durée du signal à l'état 1 est alors égale à : 20 % x 500 ms = 100 ms.

Si %I0.0.0 est réglé sur 0 et %I0.0.1 est réglé sur 1, le rapport %PWM0.R est réglé sur 50 % (durée de 250 ms).

Si %I0.0.0 et %I0.0.1 sont réglés sur 1, le rapport %PWM0.R est réglé sur 80 % (durée de 400 ms).

Exemple de programmation :



**Cas spécifiques**

Le tableau suivant présente une liste de cas spécifiques de fonctionnement du bloc fonction PWM.

Cas spécifique	Description
Effet d'un redémarrage à froid (%S0=1)	Règle le rapport %PWMi.R sur 0. En complément, la valeur de %PWMi.P est rétablie sur sa valeur configurée d'origine et prévaudra sur toute modification apportée dans l'éditeur de tables d'animation ou l'afficheur optionnel.
Effet d'un redémarrage à chaud (%S1=1)	Aucun effet.
Incidence du fait que les sorties sont dédiées au bloc %PWM	Le fait de forcer la sortie %Q0.0.0 ou %Q0.0.1 à l'aide d'un périphérique de programmation n'interrompt pas la génération du signal.

## Bloc fonction sortie du générateur d'impulsions (%PLS)

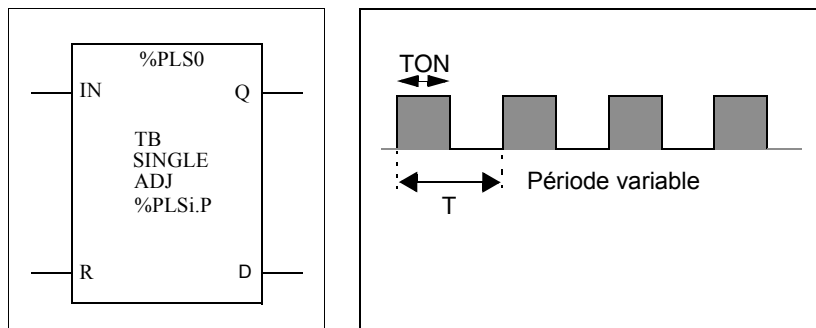
### Introduction

Le bloc fonction %PLS est utilisé pour générer des signaux carrés. Deux fonctions %PLS sont disponibles sur les voies de sorties dédiées %Q0.0.0 ou %Q0.0.1. Le bloc fonction %PLS autorise uniquement une largeur de signal unique ou un cycle d'activité de 50 %. Vous pouvez choisir de limiter le nombre d'impulsions ou le moment où le train d'impulsion est exécuté. Cela peut être déterminé au moment de la configuration et/ou de la mise à jour par l'application utilisateur.

**Note :** Les automates disposant de sorties à relais pour ces deux voies ne prennent pas en charge cette fonction %PLS.

### Représentation

Exemple de bloc fonction du générateur d'impulsions en mode mot simple :



- $TON = T/2$  pour les bases temps 0,142 ms et 0,57 ms  
 $= (\%PLSi.P * TB) / 2$
- $TON = [partie\ entière(\%PLSi.P) / 2] * TB$  pour les bases temps 10 ms à 1 s.

**Caractéristiques** Le tableau suivant présente les caractéristiques du bloc fonction PLS :

Fonction	Objet	Description
Base temps	TB	0,142 ms, 0,57 ms, 10 ms, 1 sec
Période prérégulée	%PLSi.P	<p>Les impulsions sur la sortie %PLS1 ne sont pas arrêtées lorsque %PLS1.N ou %PLS1.ND* est atteint pour les bases temps 0,142 ms et 0,57 ms.</p> <ul style="list-style-type: none"> <li>• <math>1 &lt; \%PLSi.P \leq 32767</math> pour une base temps de 10 ms ou 1 s</li> <li>• <math>0 &lt; \%PLSi.P \leq 255</math> pour une base temps de 0,57 ms ou 0,142 ms</li> <li>• 0 = Fonction non utilisée.</li> </ul> <p>Pour obtenir une précision satisfaisante du cycle d'activité avec les bases temps de 10 ms et 1 s, il est conseillé d'avoir un %PLSi <math>\geq 100</math> si P est impaire.</p>
Nombre d'impulsions	%PLSi.N %PLSi.ND *	<p>Le nombre d'impulsions à générer sur une période T peut être limité à <math>0 \leq \%PLSi.N \leq 32\,767</math> en mode standard ou à <math>0 \leq \%PLSi.ND \leq 4\,294\,967\,295</math> en mode mot double. La valeur par défaut est mise à 0.</p> <p>Pour produire un nombre illimité d'impulsions, réglez %PLSi.N ou %PLSi.ND sur zéro. Il est toujours possible de modifier le nombre d'impulsions sans tenir compte du paramétrage de l'option Réglable.</p>
Réglable	Y/N	Lorsque défini sur Y (Oui), il est possible de modifier la valeur de présélection %PLSi.P via l'IHM ou l'éditeur de tables d'animation. Lorsque défini sur N (Non), il est impossible d'accéder à cette présélection.
Entrée générateur d'impulsions	IN	A l'état 1, la génération des impulsions se fait sur la voie de sortie dédiée. A l'état 0, la voie de sortie est mise à 0.
Entrée RAZ	R	A l'état 1, les sorties %PLSi.Q et %PLSi.D sont mises à 0. Le nombre d'impulsions générées sur une période T est mis à 0.
Génération d'impulsions sur sortie courante	%PLSi.Q	A l'état 1, le signal des impulsions est généré sur la voie de sortie dédiée configurée.
Sortie de génération d'impulsions terminée	%PLSi.D	A l'état 1, la génération du signal est terminée. Le nombre voulu d'impulsions a été généré.

**Note :**

**Note :** (\*) correspond à une variable de mot double.

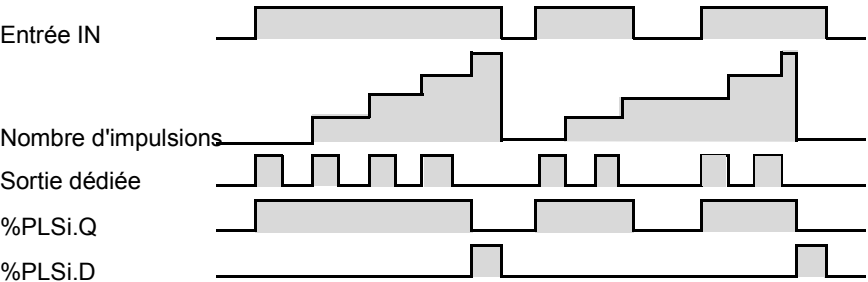
**Plage de périodes**

La valeur de présélection et la base temps peuvent être modifiées au moment de la configuration. Ces paramètres sont utilisés pour fixer la période du signal  $T = \%PLSi.P * BT$ . Plage de périodes disponible :

- 0,142 ms à 36,5 ms en pas de 0,142 ms (27,4 Hz à 7 kHz)
- 0,57 ms à 146 ms en pas de 0,57 ms (6,84 Hz à 1,75 kHz)
- 20 ms à 5,45 min en pas de 10 ms
- 2 s à 9,1 heures en pas de 1 s

**Fonctionnement**

L'exemple suivant illustre le bloc fonction %PLS.



**Cas particuliers**

<b>Cas particulier</b>	<b>Description</b>
Effet d'un redémarrage à froid (%S0=1)	Règle la fonction %PLSi.P sur la valeur définie au cours de la configuration.
Effet d'un redémarrage à chaud (%S1=1)	Aucun effet
Effet de la modification de la valeur de présélection (%PLSi.P)	Prend effet immédiatement
Incidence du fait que les sorties sont dédiées au bloc %PLS	Le fait de forcer la sortie %Q0.0.0 ou %Q0.0.1 à l'aide d'un dispositif de programmation n'interrompt pas la génération du signal.

**Note :** %PLSx.D est défini lorsque le nombre voulu d'impulsions a été généré. Il est ensuite remis à zéro en réinitialisant les entrées IN ou R sur 1.

## Bloc fonction programmeur cyclique (%DR)

---

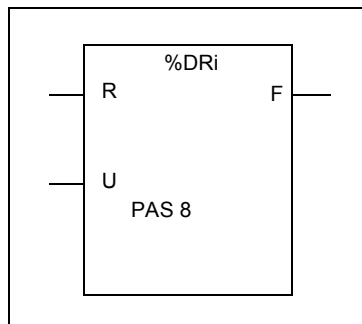
### Introduction

Le fonctionnement des programmeurs cycliques est semblable à celui des programmeurs cycliques électromécaniques qui permettent la modification de pas en fonction d'événements externes. A chaque pas, le point haut d'une came donne une commande exécutée par l'automatisme. Dans le cas d'un programmeur cyclique, ces points hauts sont symbolisés par l'état 1 pour chacun des pas et sont affectés aux bits de sortie %Qi.j ou aux bits internes %Mi, appelés "bits de contrôle".

---

### Illustration

L'exemple suivant illustre l'utilisation du bloc fonction programmeur cyclique.



Bloc fonction programmeur cyclique

---



**Paramètres**

Le bloc fonction programmeur cyclique possède les paramètres suivants.

Paramètre	Etiquette	Valeur
Numéro	%DRi	0 à 3 automates compacts 0 à 7 automates modulaires
Numéro du pas courant	%DRi.S	0<%DRi.S<7. Mot pouvant être lu et écrit. La valeur écrite doit être une valeur décimale immédiate. Une fois écrite, la valeur sera prise en compte à la prochaine exécution du bloc fonction.
Nombre de pas		1 à 8 (par défaut)
Entrée retour au pas 0(ou à l'instruction)	R (Reset)	A l'état 1, règle le programmeur cyclique sur le pas 0.
Entrée (ou instruction) avancée	U (haut)	Sur un front montant, provoque le passage du programmeur cyclique au pas suivant et met à jour les bits de contrôle.
Sortie	F (plein)	Indique que le pas courant est égal au dernier pas défini. Le bit associé %DRi.F peut être testé (par exemple, %DRi.F=1, si %DRi.S= nombre de pas configurés - 1).
Bits de contrôle		Bits de sortie ou bits internes associés au pas (16 bits de contrôle) et définis dans l'éditeur de configuration.

## Fonctionnement du bloc fonction programmeur cyclique %DRi

### Introduction

Le programmeur cyclique comprend :

- Une matrice de données constantes (des cames), organisée en huit pas (numérotés de 0 à 7) et 16 bits de données (état du pas), disposés en colonnes numérotées de 0 à F.
- Une liste des bits de contrôle est associée à une sortie configurée (%Qi.j.k) ou à un mot mémoire (%Mi). Au cours du pas courant, les bits de contrôle prennent les états binaires définis pour ce pas.

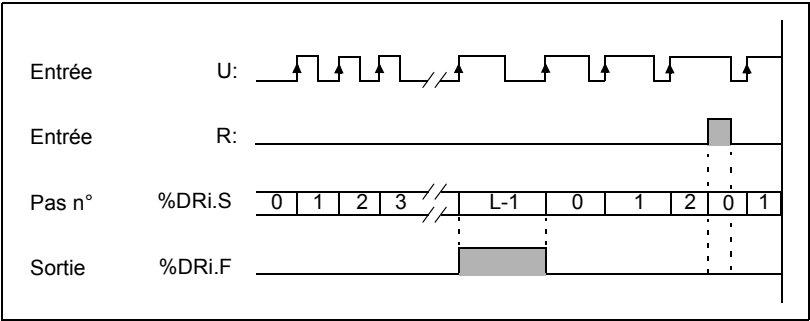
L'exemple présenté dans le tableau suivant résume les caractéristiques principales du programmeur cyclique.

Colonne	0	1	2		D	E	F
Bits de contrôle	%Q0.1	%Q0.3	%Q1.5		%Q0.6	%Q0.5	%Q1.0
Pas 0	0	0	1		1	1	0
Pas 1	1	0	1		1	0	0
Pas 5	1	1	1		0	0	0
Pas 6	0	1	1		0	1	0
Pas 7	1	1	1		1	0	0

### Fonctionnement

Dans l'exemple précédent, le pas 5 est le pas courant, les bits de contrôle %Q0.1, %Q0.3 et %Q1.5 sont à l'état 1 ; les bits de contrôle %Q0.6, %Q0.5 et %Q1.0 sont à l'état 0. Le numéro du pas courant est incrémenté d'une unité sur chaque front montant de l'entrée U (ou lors de l'activation de l'instruction U). Le pas courant peut être modifié par le programme.

**Chronogramme**      Le chronogramme suivant illustre le fonctionnement du programmeur cyclique.



**Cas particuliers**      Le tableau suivant présente une liste des cas spécifiques de fonctionnement du programmeur cyclique.

Cas spécifique	Description
Effets d'un redémarrage à froid (%S0=1)	Provoque la réinitialisation du programmeur cyclique au pas 0 (mise à jour des bits de contrôle).
Effet d'une reprise à chaud (%S1=1)	Met à jour les bits de contrôle d'après le pas courant.
Effet d'un saut de programme	Le fait de ne plus scruter le programmeur cyclique ne remet pas les bits de contrôle à zéro.
Mise à jour des bits de contrôle	Survient uniquement en cas de changement de pas ou lors d'un démarrage à froid ou d'un redémarrage à chaud.

## Programmation et configuration des programmeurs cycliques

### Introduction

Dans l'exemple suivant de programmation et de configuration d'un programmeur cyclique, les six premières sorties (%Q0.0 à %Q0.5) sont activées les unes à la suite des autres, chaque fois que l'entrée %I0.1 est mise à 1. L'entrée I0.0 remet les sorties à zéro.

### Exemple de programmation

L'illustration suivante représente un bloc fonction programmeur cyclique et présente des exemples de programmation réversible et non réversible.

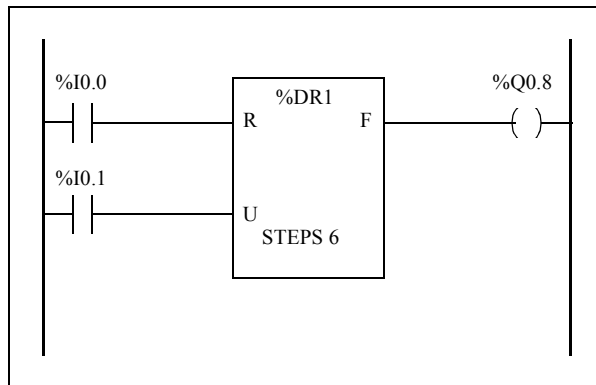


Schéma à contacts

```
BLK    %DR1
LD     %I0.0
R
LD     %I0.1
U
OUT_BLK
LD     F
ST     %Q0.8
END_BLK
```

**Configuration**

Les informations suivantes sont définies au moment de la configuration :

- nombre de pas : 6
- états de sortie (bits de contrôle) pour chaque pas du programmeur cyclique

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Etape 1 :	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Etape 2 :	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Etape 3 :	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Etape 4 :	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Etape 5 :	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Etape 6 :	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

- affectation des bits de contrôle

1 :	%Q0.0	4 :	%Q0.1
2 :	%Q0.2	5 :	%Q0.3
3 :	%Q0.4	6 :	%Q0.5

## Bloc fonction compteur rapide (%FC)

---

### Introduction

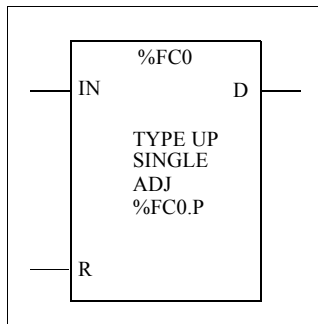
Le bloc fonction compteur rapide (%FC) sert à la fois de compteur et de décompteur. Il peut compter le front montant des entrées TOR pour des fréquences allant jusqu'à 5 kHz en mode de calcul mot simple ou mot double. Etant donné que les compteurs rapides (FC) sont gérés par des interruptions matérielles spécifiques, le maintien du taux d'échantillonnage maximal des fréquences peut varier en fonction de la configuration de votre application et de votre matériel.

Les automates compacts TWDLCA•40DRF peuvent contenir jusqu'à quatre compteurs rapides alors que toutes les autres gammes d'automates compacts ne peuvent être configurées que pour utiliser au maximum trois compteurs rapides. Quant aux automates modulaires, ils ne peuvent en comporter que deux. Les blocs fonction compteur rapide %FC0, %FC1, %FC2 et %FC3 utilisent respectivement les entrées dédiées %I0.0.2, %I0.0.3, %I0.0.4 et %I0.0.5. Ces bits ne sont pas exclusivement réservés à ces blocs fonction. L'affectation de ces bits doit être déterminée selon l'utilisation de ces ressources dédiées par d'autres blocs fonction.

---

### Illustration

L'illustration suivante présente un exemple de bloc fonction compteur rapide (FC) en mode mot simple.



**Paramètres**

Le tableau suivant présente les différents paramètres du bloc fonction compteur rapide (FC).

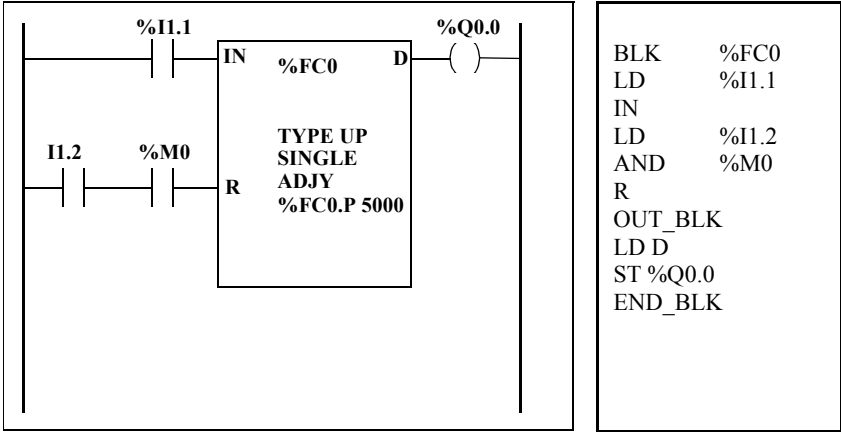
Paramètre	Etiquette	Description
Fonction	TYPE	Paramètre défini lors de la configuration et permettant de choisir entre le compteur et le décompteur.
Valeur de présélection	%FCi.P %FCi.PD	Valeur initiale définie : ->entre 1 et 65 635 en mode standard, ->entre 1 et 4 294 967 295 en mode mot double.
Réglable	O/N	Lorsqu'il est défini sur O, il est possible de modifier la valeur de présélection %FCi.P ou %FCi.PD et la valeur courante %FCi.V ou %FCi.VD à l'aide de l'afficheur ou de l'éditeur de tables d'animation. Lorsqu'il est défini sur N, il n'est pas possible d'accéder à cette présélection.
Valeur courante	%FCi.V %FCi.VD	La valeur courante évolue de manière croissante ou décroissante selon la fonction sélectionnée (comptage ou décomptage). Pour le comptage, la valeur de comptage courante est mise à jour. Elle peut atteindre 65 535 en mode standard (%FCi.V) et 4 294 967 295 en mode mot double (%FCi.VD). Pour le décomptage, la valeur courante est la valeur de présélection %FCi.P ou %FCi.PD. Elle peut décroître jusqu'à zéro.
Entrer pour valider	IN	A l'état 1, la valeur courante est mise à jour selon les impulsions appliquées à l'entrée physique. A l'état 0, la valeur courante reste inchangée.
Remise à zéro	%FCi.R	Paramètre utilisé pour initialiser le bloc. A l'état 1, la valeur courante est remise à 0 lorsque le bloc est configuré en tant que compteur, ou définie sur %FCi.P ou %FCi.PD lorsqu'il est configuré en tant que décompteur. Le bit Terminé %FCi.D reprend sa valeur par défaut.
Terminé	%FCi.D	Ce bit est réglé sur 1 lorsque %FCi.V ou %FCi.VD atteint %FCi.P ou %FCi.PD (bloc configuré en tant que compteur) ou lorsque %FCi.V ou %FCi.VD atteint zéro (bloc configuré en tant que décompteur). Ce bit en lecture seule est remis à 0 uniquement lorsque le paramètre %FCi.R est réglé sur 1.

**Remarque** Lorsque le bloc est configuré comme réglable, l'application peut modifier la valeur de présélection %FCi.P ou %FCi.PD et la valeur courante %FCi.V ou %FCi.VD à tout moment. Cependant, une nouvelle valeur est prise en compte uniquement lorsque la réinitialisation de l'entrée est active ou sur le front montant de la sortie %FCi.D. Cela permet d'effectuer plusieurs comptages successifs sans perdre une seule impulsion.

**Fonctionnement** Lorsque le bloc est configuré comme compteur, la valeur courante est incrémentée de 1 dès qu'un front montant apparaît au niveau de l'entrée dédiée. Lorsque la valeur de présélection %FCi.P ou %FCi.PD est atteinte, le bit de sortie Terminé %FCi.D est défini sur 1 et la valeur courante %FCi.V ou %FCi.VD devient égale à zéro.

Lorsque le bloc est configuré comme décompteur, la valeur courante est diminuée de 1 dès qu'un front montant apparaît au niveau de l'entrée dédiée. Lorsque la valeur est zéro, le bit de sortie Terminé %FCi.D est défini sur 1 et la valeur courante %FCi.V ou %FCi.VD devient égale à la valeur de présélection.

**Configuration et programmation** Dans l'exemple ci-dessous, l'application compte un nombre d'éléments (5 000 maximum) pendant que %I1.1 est défini sur 1. L'entrée pour %FC0 est l'entrée dédiée %IO.0.2. Lorsque la valeur de présélection est atteinte, %FC0.D est défini sur 1 et conserve la même valeur jusqu'à ce que %FC0.R soit commandé par le résultat de l'opération booléenne "AND" sur %I1.2 et %M0.





**Cas particuliers** Le tableau suivant présente une liste de cas spécifiques de fonctionnement du bloc fonction %FC.

Cas spécifique	Description
Effet d'un démarrage à froid (%S0=1)	Réinitialise les attributs %FC sur les valeurs configurées par l'utilisateur ou l'application utilisateur.
Effet d'un redémarrage à chaud (%S1=1)	Aucun effet.
Effet d'un arrêt de l'automate	%FC continue à compter selon les paramètres activés au moment de l'arrêt de l'automate.

## Bloc fonction compteur rapide (%VFC)

---

### Introduction

Le bloc fonction compteur rapide (%VFC) peut être configuré à l'aide de TwidoSoft pour exécuter l'une des fonctions suivantes :

- Compteur/décompteur
- Compteur/décompteur bi-phases
- Compteur simple
- Décompteur simple
- Fréquencemètre

Le bloc %VFC prend en charge le comptage des entrées TOR pour des fréquences allant jusqu'à 20 kHz en mode de calcul mot simple ou mot double. Les automates compacts TWDLCA•40DRF peuvent contenir jusqu'à deux compteurs rapides (VFC) alors que les autres gammes d'automates compacts ne peuvent en comporter qu'un seul. Quant aux automates modulaires, ils peuvent en configurer jusqu'à deux.

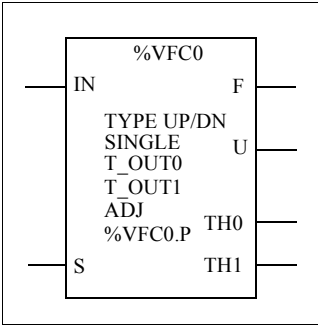
---

**Affectations des E/S dédiées**

Les blocs fonction compteur rapide (%VFC) utilisent des entrées dédiées et des entrées et sorties auxiliaires. Ces entrées et ces sorties ne sont pas exclusivement réservées à ces blocs fonction. Leur affectation doit être déterminée selon l'utilisation de ces ressources dédiées par d'autres blocs fonction. Le tableau ci-après récapitule les affectations :

		Entrées principales		Entrées auxiliaires		Sorties réflexes	
%VFC0	Utilisation choisie	Entrée IA	Entrée IB	IPres	Ica	Sortie 0	Sortie 1
	Compteur/décompteur	%I0.0.1	%I0.0.0 (CO = 0/DE = 1)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	Compteur/Décompteur bi-phases	%I0.0.1	%I0.0.0 (Impulsion)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	Compteur simple	%I0.0.1	(2)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	Décompteur simple	%I0.0.1	(2)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	Fréquencemètre	%I0.0.1	(2)	(2)	(2)	(2)	(2)
%VFC1	Utilisation choisie	Entrée IA	Entrée IB)	IPres	Ica	Sortie 0	Sortie 1
	Compteur/décompteur	%I0.0.7	%I0.0.6 (CO = 0/DE = 1)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	Compteur/Décompteur bi-phases	%I0.0.7	%I0.0.6 (Impulsion)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	Compteur simple	%I0.0.7	(2)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	Décompteur simple	%I0.0.7	(2)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	Fréquencemètre	%I0.0.7	(2)	(2)	(2)	(2)	(2)
<b>Commentaires :</b> (1) = facultatif (2) = non utilisé <b>IPres</b> = entrée de présélection <b>Ica</b> = entrée de capture Lorsqu'elle n'est pas utilisée, l'entrée ou la sortie reste une E/S TOR normale gérée par l'application au cours du cycle principal.  Si %I0.0.2 est utilisé, %FC0 n'est pas disponible. Si %I0.0.3 est utilisé, %FC2 n'est pas disponible. Si %I0.0.4 est utilisé, %FC3 n'est pas disponible.							

**Illustration** La figure suivante représente le compteur rapide (%VFC) en mode mot simple :



**Caractéristiques** Le tableau suivant répertorie les caractéristiques du bloc fonction compteur rapide (%VFC).

Fonction	Description	Valeurs	Utilisa- tion du bloc %VFC	Accès en cours d'exécu- tion
Valeur courante (%VFCi.V) (%VFCi.VD*)	La valeur courante est augmentée ou diminuée en fonction des entrées physiques et de la fonction sélectionnée. Cette valeur peut-être présélectionnée ou initialisée à l'aide de l'entrée de présélection (%VFCi.S).	%VFCi.V : 0 -> 65 535 %VFCi.VD : 0 -> 4 294 967 295	CM	Lecture
Valeur de présélection (%VFCi.P) (%VFCi.PD*)	Uniquement utilisée par la fonction compteur/décompteur et par le comptage ou le décomptage simple.	%VFCi.P : 0 -> 65 535 %VFCi.PD : 0 -> 4 294 967 295	CM ou FM	Lecture et écriture (1)
Valeur de capture (%VFCi.C) (%VFCi.CD*)	Uniquement utilisée par la fonction de comptage/décomptage et par le comptage/décomptage simple.	%VFCi.C : 0 -> 65 535 %VFCi.CD : 0 -> 4 294 967 295	CM	Lecture

Fonction	Description	Valeurs	Utilisation du bloc %VFC	Accès en cours d'exécution
Sens de comptage (%VFCi.U)	Défini par le système, ce bit est utilisé par la fonction de comptage/décomptage pour indiquer le sens de comptage : Pour un compteur/décompteur simple, %I0.0.0 détermine le sens de %VFC0, et %I0.0.6 détermine le sens de %VFC1. Pour un compteur/décompteur bi-phases, la différence de phase entre les deux signaux détermine le sens de comptage. Pour %VFC0, %I0.0 est dédié à IB et %I0.1 à IA. Pour %VFC1, %I0.6 est dédié à IB et %I0.7 à IA.	0 (Décomptage) 1 (Comptage)	CM	Lecture
Activation de la sortie réflexe 0 (%VFCi.R)	Validation de la sortie réflexe 0	0 (Désactivé) 1 (Activé)	CM	Lecture et écriture (2)
Activer sortie réflexe 1 (%VFCi.S)	Validation sortie réflexe 1	0 (Désactivé) 1 (Activé)	CM	Lecture et écriture (2)
Valeur seuil S0 (%VFCi.S0) %VFCi.S0D	Ce mot contient la valeur de seuil 0. La signification est définie au moment de la configuration du bloc fonction. Remarque : Cette valeur doit être inférieure à %VFCi.S1.	%VFCi.S0 : 0 -> 65 535 %VFCi.S0D : 0 -> 4 294 967 295	CM	Lecture et écriture (1)
Valeur seuil S1 (%VFCi.S1) (%VFCi.S1D*)	Ce mot contient la valeur seuil 0. La signification est définie au moment de la configuration du bloc fonction. Remarque : Cette valeur doit être supérieure à %VFCi.S0.	%VFCi.S1 : 0 -> 65 535 %VFCi.S1D : 0 -> 4 294 967 295	CM	Lecture et écriture (1)
Base temps de la mesure de fréquence (%VFCi.T)	Elément de configuration de la base temps (100 ou 1 000 millisecondes).	1 000 ou 100	MF	Lecture et écriture (1)
Réglable (Y/N)	Elément de configuration qui, lorsqu'il est sélectionné, permet à l'utilisateur de modifier les valeurs de présélection, de seuil et de base temps de la mesure de fréquence en cours d'exécution.	N (Non) Y (Oui)	CM ou FM	Non
Entrée pour valider (IN)	Utilisée pour valider ou inhiber la fonction courante.	0 (Non)	CM ou FM	Lecture et écriture (3)

Fonction	Description	Valeurs	Utilisation du bloc %VFC	Accès en cours d'exécution
Entrée de présélection (S)	Dépend de la configuration à l'état 1 : <ul style="list-style-type: none"> <li>comptage/décomptage ou décomptage : initialise la valeur courante avec la valeur de présélection.</li> <li>comptage simple : remet la valeur courante à 0.</li> </ul> Cette fonction permet également d'initialiser la commande des sorties seuils et prend en compte toutes les modifications apportées par un utilisateur aux valeurs seuils définies par l'afficheur ou le programme utilisateur.	0 ou 1	MC ou MF	Lecture et écriture
Sortie débordement (F)	de 0 à 65 535 ou de 65 535 à 0 en mode standard de 0 à 4 294 967 295 ou de 4 294 967 295 à 0 en mode mot double	0 ou 1	CM	Lecture
Seuil Bit 0 (%VFCi.TH0)	A l'état 1 lorsque la valeur courante est supérieure ou égale à la valeur seuil %VFCi.S0. Nous conseillons de tester ce bit une seule fois dans le programme, car il est mis à jour en temps réel. L'application utilisateur est responsable de la validité de la valeur au moment de son utilisation.	0 ou 1	CM	Lecture
Seuil Bit 1 (%VFCi.TH1)	A l'état 1 lorsque la valeur courante est supérieure ou égale à la valeur seuil %VFCi.S1. Nous conseillons de tester ce bit une seule fois dans le programme, car il est mis à jour en temps réel. L'application utilisateur est responsable de la validité de la valeur au moment de son utilisation.	0 ou 1	MC	Lecture

(\*)Correspond à une variable de double mot 32 bits. L'option de double mot est disponible sur tous les automates à l'exception des automates Twido TWDLC•A10DRF.

(1) Accessible en écriture uniquement si la fonction Réglable est réglée sur un.

(2) Accès disponible si configuré uniquement.

(3)Accès en lecture et en écriture seulement à partir de l'application. Accès impossible à partir de l'afficheur ou de l'éditeur de tables d'animation.

MC = Mode Comptage

MF = Mode Fréquence

**Description de la fonction de comptage** La fonction de comptage rapide (%VFC) fonctionne à une fréquence maximale de 20 kHz et pour une plage de valeurs allant de 0 à 65 535 (en mode standard) et de 0 à 4 294 967 295. Les impulsions à compter sont appliquées de la manière suivante :

Tableau :

Fonction	Description	%VFC0		%VFC1	
		IA	IB	IA	IB
Compteur/ Décompteur	Les impulsions sont appliquées à l'entrée physique ; l'opération courante (comptage/décomptage) est définie par l'état de l'entrée physique IB.	%I0.0.1	%I0.0.0	%I0.0.7	%I0.0.6
Compteur/ Décompteur bi- phases	Les deux phases du codeur sont appliquées aux entrées physiques IA et IB.	%I0.0.1	%I0.0.0	%I0.0.7	%I0.0.6
Compteur simple	Les impulsions sont appliquées à l'entrée physique IA. IB n'est pas utilisée.	%I0.0.1	ND	%I0.0.7	ND
Décompteur simple	Les impulsions sont appliquées à l'entrée physique IA. IB n'est pas utilisée.	%I0.0.1	ND	%I0.0.7	N/A

### Remarques sur les blocs fonction

Les opérations de comptage ou de décomptage sont effectuées sur le front montant des impulsions et ce, uniquement lorsque le bloc compteur est activé.

Deux entrées facultatives sont utilisées en mode de comptage : ICa et IPres. ICa est utilisée pour capturer la valeur courante (%VFCi.V ou %VFCi.VD) et la stocker dans %VFCi.C ou %VFCi.CD. Les entrées ICa sont définies sur %I0.0.3 pour %VFC0 et sur %I0.0.4 pour %VFC1, le cas échéant.

Lorsque l'entrée IPres est active, la valeur courante est affectée de la manière suivante :

- Pour le comptage, %VFCi.V ou %VFCi.VD sont remis à 0.
- Pour le décomptage, %VFCi.V ou %VFCi.VD sont écrits respectivement avec le contenu de %VFCi.P ou %VFCi.PD.
- Pour le comptage de fréquence, %VFCi.V ou %VFCi.PD sont mis à 0.

Avertissement : %VFCi.F est également mis à 0. Les entrées IPres sont définies sur %I0.0.2 pour %VFC0 et sur %I0.0.5 pour %VFC1, le cas échéant.

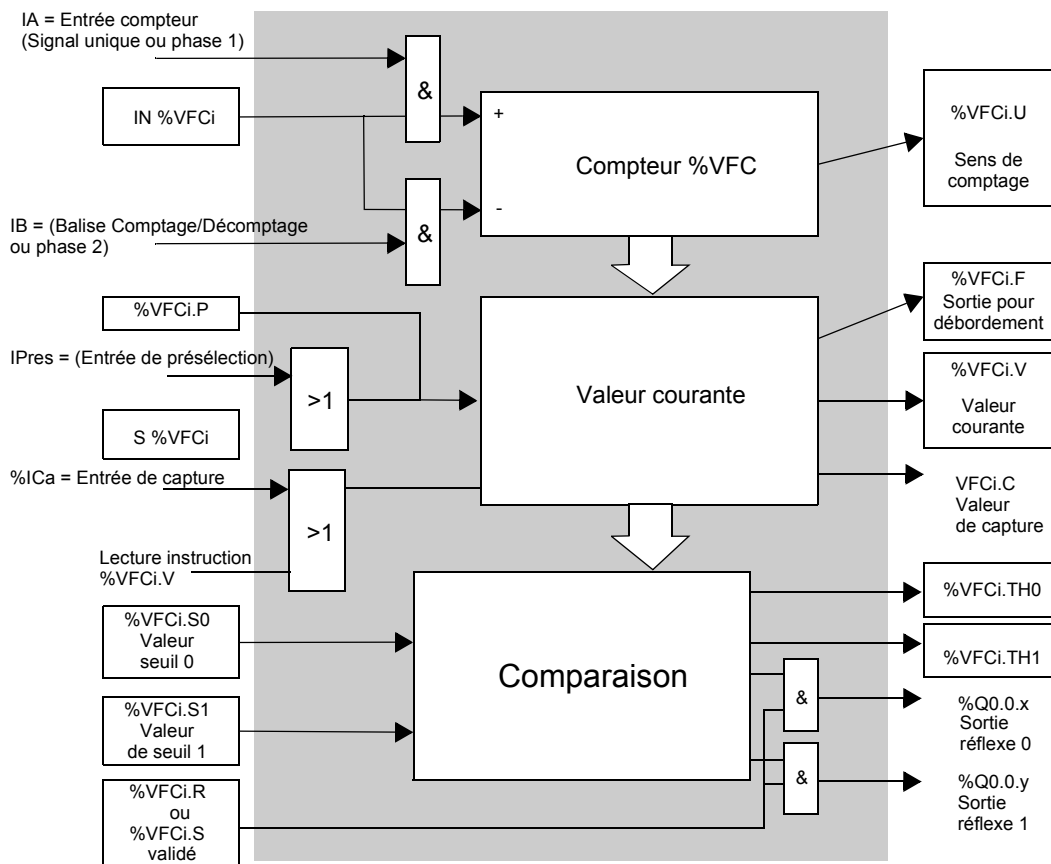
### Remarques sur les sorties des blocs fonction

Pour toutes les fonctions, les valeurs courantes sont comparées aux deux seuils (%VFCi.S0 ou %VFCi.S0D et %VFCi.S1 ou %VFCi.S1D). Les deux objets bits (%VFCi.TH0 et %VFCi.TH1) sont fonction des résultats de cette comparaison. C'est-à-dire qu'ils sont réglés sur 1 lorsque la valeur courante est supérieure ou égale au seuil correspondant ou remis à 0 dans le cas contraire. Les sorties réflexes (si elles sont configurées) sont réglées sur 1 en fonction de ces comparaisons. Remarque : Aucune, une ou deux sorties peuvent être configurées.

%VFC.U est une sortie du bloc fonction. Elle indique le sens de variation du compteur (1 pour comptage, 0 pour décomptage).

### Schéma de la fonction de comptage

L'illustration suivante représente un schéma de fonction de comptage en mode standard (en mode double mot, vous utiliserez en conséquence les variables de la fonction de doubles mots) :



**Note :** Les sorties sont gérées indépendamment du temps de cycle automate. Le temps de réponse est compris entre 0 et 1 ms.

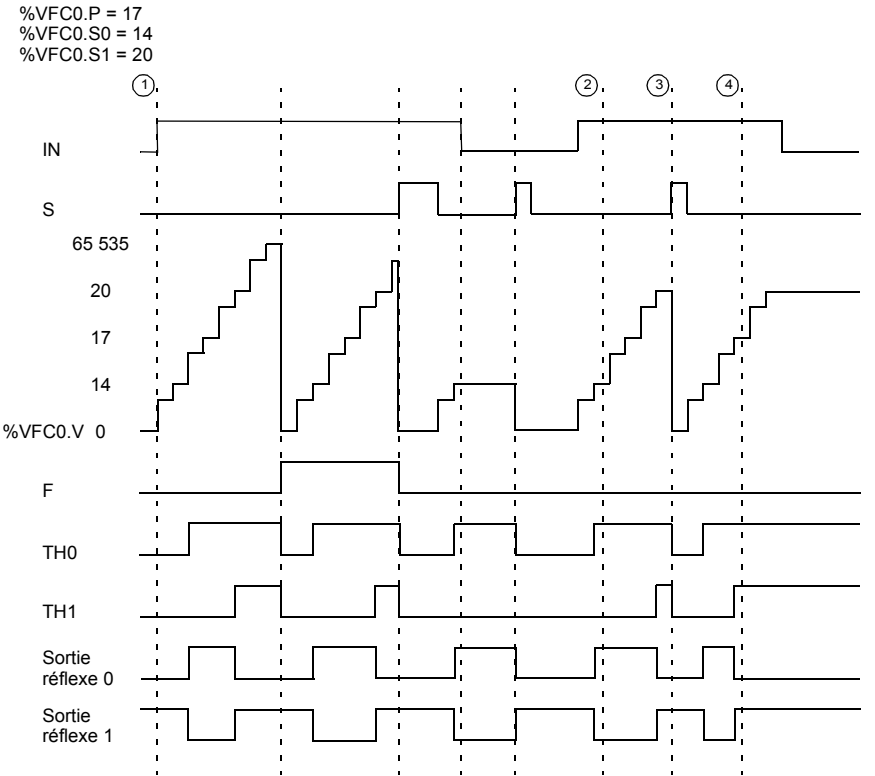


Opération de  
comptage simple

Voici un exemple de l'utilisation de %VFC en mode comptage simple. Les éléments de configuration suivants ont été définis pour cet exemple :  
La valeur de présélection, %VFC0.P, est égale à 17. Le seuil inférieur, %VFC0.S0, est égal à 14 et le seuil supérieur, %VFC0.S1, à 20.

Sortie réflexe	<%VFC.S0	%VFC0.S0 <==< %VFC0.S1	>= %VFC0.S1
%Q0.0.2		X	
%Q0.0.3	X		X

Exemple de chronogramme :



- ① : %VFC0.U = 1 car %VFC est un compteur
- ② : modification de %VFC0.S1 sur 17
- ③ : l'activation de l'entrée S permet d'accorder la nouvelle valeur du seuil S1 lors du comptage suivant
- ④ : une interception de la valeur courante a lieu, ainsi, %VFC0.C = 17

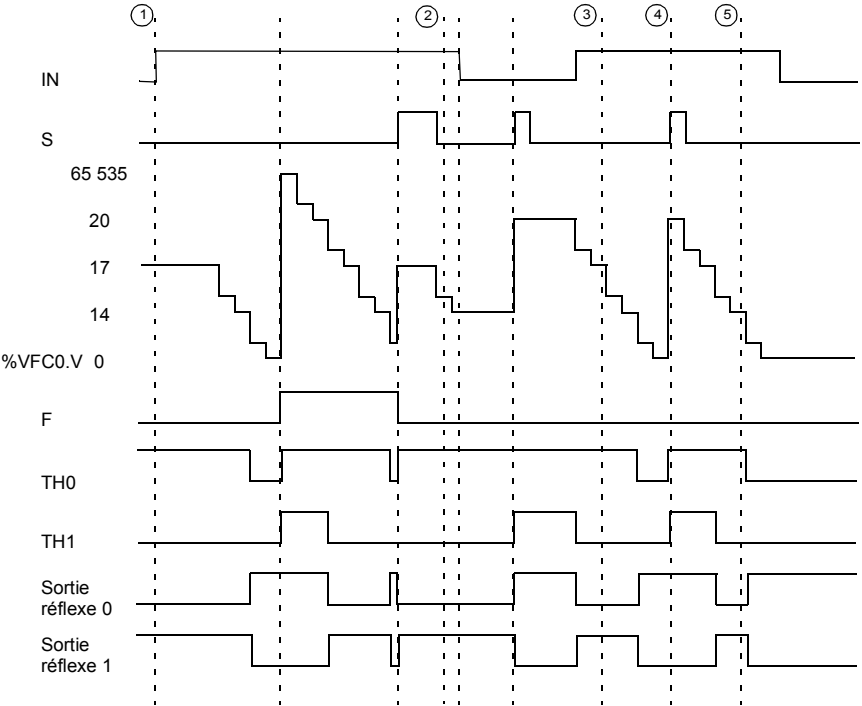
Opération de décomptage simple

Voici un exemple de l'utilisation de %VFC en mode décomptage simple. Les éléments de configuration suivants ont été définis pour cet exemple : La valeur de présélection, %VFC0.P, est égale à 17. Le seuil inférieur, %VFC0.S0, est égal à 14 et le seuil supérieur, %VFC0.S1, à 20.

Sortie réflexe	<%VFC.S0	%VFC0.S0 <=< %VFC0.S1	>= %VFC0.S1
%Q0.0.2	X		X
%Q0.0.3		X	

Exemple :

%VFC0.P = 17  
%VFC0.S0 = 14  
%VFC0.S1 = 20



- ① : %VFC0.U = 0 car %VFC est un décompteur
- ② : modification de %VFC0.P sur 20
- ③ : modification de %VFC0.S1 sur 17
- ④ : L'activation de l'entrée S permet d'accorder la nouvelle valeur du seuil S1 lors du décompte suivant
- ⑤ : une capture de la valeur courante a lieu, ainsi, %VFC0.C = 17

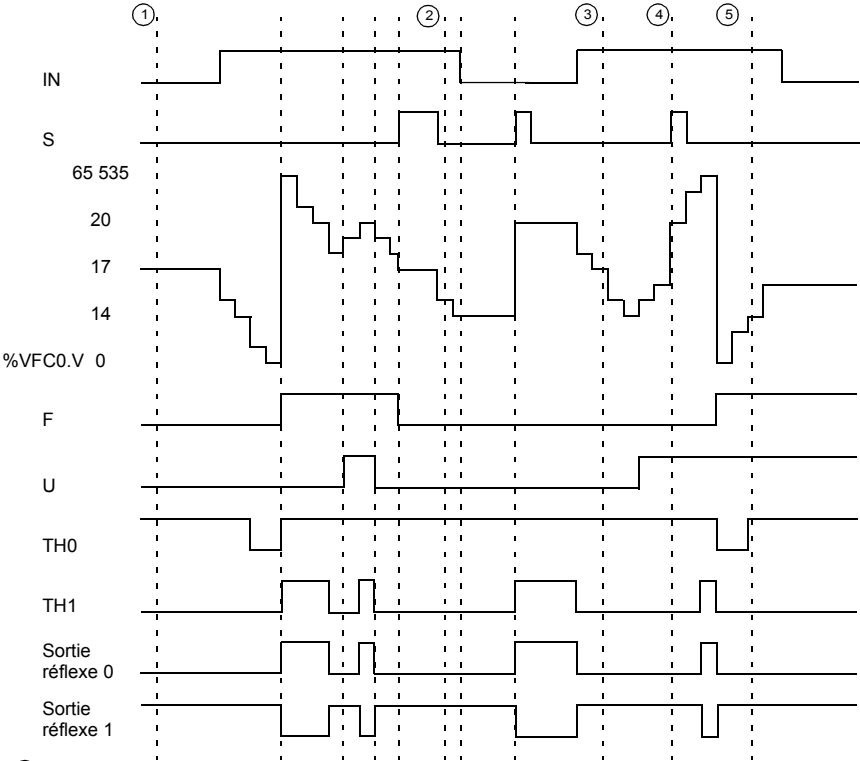
Opération de  
comptage/  
décomptage

Voici un exemple de l'utilisation de %VFC en mode comptage/décomptage. Les éléments de configuration suivants ont été définis pour cet exemple :  
La valeur de présélection, %VFC0.P, est égale à 17. Le seuil inférieur, %VFC0.S0, est égal à 14 et le seuil supérieur, %VFC0.S1, à 20.

Sortie réflexe	<%VFC.S0	%VFC0.S0 <= %VFC0.S1	%VFC0.S1
%Q0.0.2			X
%Q0.0.3	X	X	

Exemple :

%VFC0.P = 17  
%VFC0.S0 = 14  
%VFC0.S1 = 20



- ① : entrées IN et S mises à 1
- ② : modification de %VFC0.P sur 20
- ③ : modification de %VFC0.S1 sur 17
- ④ : L'activation de l'entrée S permet d'accorder la nouvelle valeur du seuil S1 lors du décompte suivant
- ⑤ : une capture de la valeur courante a lieu, ainsi, %VFC0.C = 17

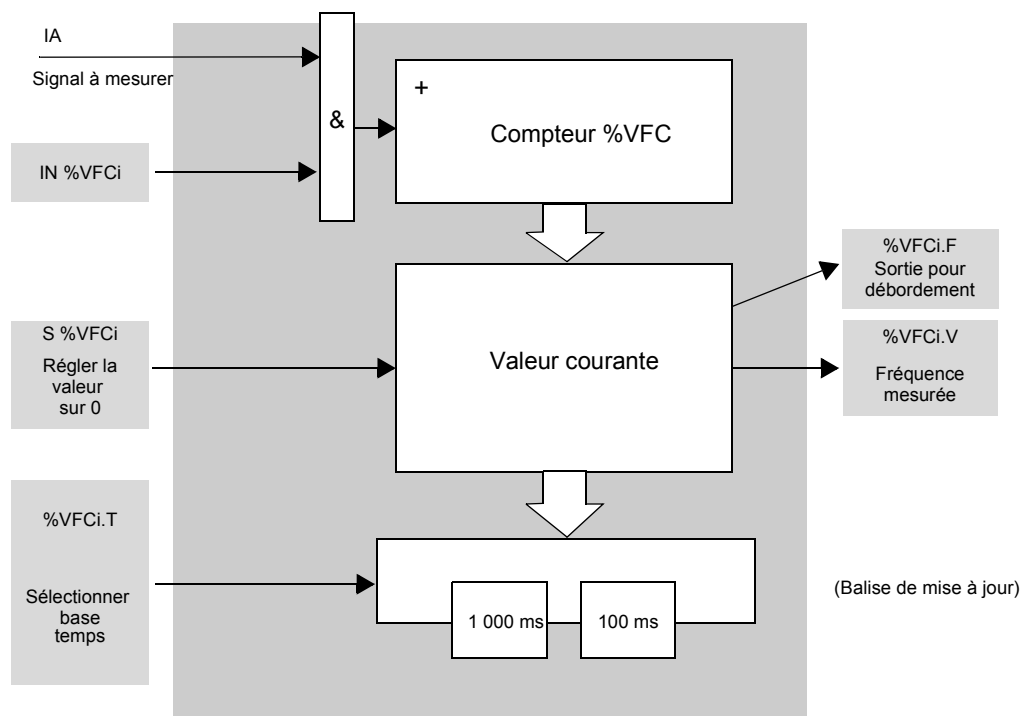
**Description de la fonction**  
**Fréquencemètre**

La fonction Fréquencemètre d'un %VFC est utilisée pour mesurer la fréquence en Hz d'un signal périodique sur l'entrée IA. La plage de fréquences pouvant être mesurées s'étend de 10 à 20 kHz. L'utilisateur peut choisir entre deux bases temps. Ce choix est effectué via un nouvel objet %VFC.T (Base temps). Une valeur de 100 correspond à une base temps de 100 ms et une valeur de 1 000 correspond à une base temps d'une seconde.

Base temps	Plage de mesure	Précision	Mise à jour
100 ms	100 Hz à 20 kHz	0,05 % pour 20 kHz, 10 % pour 100 Hz	10 fois par seconde
1 s	10 Hz à 20 kHz	0,005 % pour 20 kHz, 10 % pour 10 Hz	Une fois par seconde

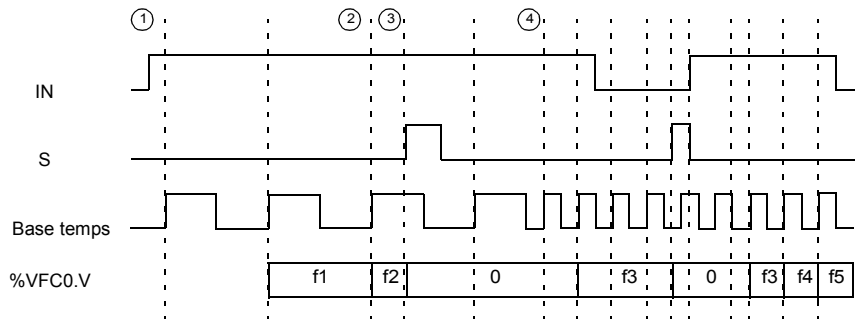
**Schéma de la  
fonction  
Fréquence**

Exemple de schéma de fonction Fréquence :



Opération  
Fréquencemètre

Voici un exemple de chronogramme de l'utilisation de %VFC en mode Fréquencemètre :



Cas particuliers

Le tableau suivant présente une liste de cas spécifiques de fonctionnement du bloc fonction %VFC.

Cas spécifique	Description
Effet d'un démarrage à froid (%S0=1)	Utilise les valeurs configurées par l'utilisateur ou par l'application utilisateur pour régler tous les attributs %VFC.
Effet d'un redémarrage à chaud (%S1=1)	Aucun effet
Effet de l'arrêt de l'automate	Le %VFC s'arrête et les sorties maintiennent leur état courant.

## Emission/réception de messages - Instruction d'échange (EXCH)

### Introduction

Il est possible de configurer un automate Twido afin qu'il puisse communiquer avec des périphériques esclaves Modbus ou envoyer et/ou recevoir des messages en mode ASCII (mode caractères).

TwidoSoft propose les fonctions suivantes pour ces communications :

- Instruction EXCH pour l'émission/la réception de messages
- Bloc fonction de contrôle d'échange %MSG assurant le contrôle des échanges de données

L'automate Twido utilise le protocole configuré pour le port spécifié lors du traitement d'une instruction EXCH. Il est possible d'affecter un protocole différent à chaque port de communication. Pour accéder aux ports de communication, ajoutez le numéro de port à la fonction EXCH ou %MSG (EXCH1, EXCH2, %MSG1, %MSG2).

De plus, les automates TWDLCAE40DRF implémentent la messagerie Modbus TCP sur le réseau Ethernet à l'aide de l'instruction EXCH3 et de la fonction %MSG3.

### Instruction EXCH

L'instruction EXCH permet à un automate Twido d'envoyer et/ou recevoir des informations vers/depuis des périphériques ASCII. L'utilisateur définit une table de mots (%MWi:L) contenant les données à envoyer et/ou recevoir (jusqu'à 250 octets de données en émission et/ou réception). Le format des tables de mots fait l'objet d'une description dans les sections relatives à chaque protocole. Un échange de message est exécuté à l'aide de l'instruction EXCH.

### Syntaxe

La syntaxe à utiliser pour l'instruction EXCH est la suivante :

[EXCHx %MWi:L]

Où : x = numéro de port série (1 ou 2), x = port Ethernet (3), L = nombre total de mots de la table de mots (121 maximum). Les valeurs contenues dans la table de mots interne %MWi:L prennent la forme i+L <= 255.

L'automate Twido doit terminer l'échange ordonné par la première instruction EXCHx avant qu'une nouvelle instruction d'échange puisse être lancée. Le bloc fonction %MSG doit être utilisé lors de l'envoi de plusieurs messages.

**Note :** Pour plus d'informations sur l'instruction EXCH3 de messagerie Modbus TCP, voir *Messagerie Modbus TCP*, p. 190.

## Bloc fonction de contrôle d'échange (%MSGx)

### Introduction

**Note :** Le "x" de %MSGx désigne le port de l'automate : "x = 1 ou 2"

- x = 1 ou 2 correspond respectivement au port série 1 ou 2 de l'automate ;
- X = 3 correspond au port réseau Ethernet de l'automate (sur l'automate TWDLCAE40DRF uniquement). Pour plus d'informations sur la fonction %MSG3, voir *Messagerie Modbus TCP*, p. 190.

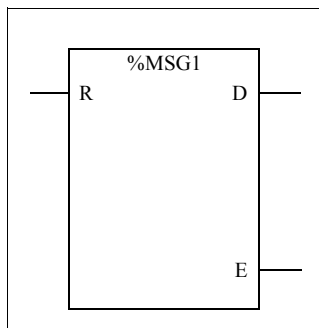
Le bloc fonction %MSGx assure la gestion des échanges de données. Ce bloc a trois fonctions :

- Vérification des erreurs de communication  
Cette fonction a pour but de s'assurer que la longueur du bloc (table de mots) programmée avec l'instruction EXCH est suffisante pour le stockage du message à envoyer (comparaison de la longueur programmée dans l'octet de poids faible du premier mot de la table de mots).
- Coordination de plusieurs messages  
Afin d'assurer la coordination de l'envoi de plusieurs messages, le bloc fonction %MSGx contient des informations permettant de s'assurer que l'émission de chaque message est bien terminée.
- Emission de messages prioritaires  
Le bloc fonction %MSGx vous permet de suspendre l'émission d'un message afin d'envoyer un message plus urgent.

La programmation du bloc fonction %MSGx est facultative.

### Illustration

L'exemple suivant illustre le bloc fonction %MSGx.





**Paramètres**

Le tableau suivant présente les différents paramètres du bloc fonction %MSGx.

Paramètre	Etiquette	Valeur
Entrée (ou instruction) RAZ	R	A l'état 1, réinitialise la communication : %MSGx.E = 0 et %MSGx.D = 1.
Sortie Communication terminée	%MSGx.D	A l'état 1, la communication est terminée si : <ul style="list-style-type: none"> <li>● fin d'émission (si émission)</li> <li>● fin de réception (réception du caractère de fin)</li> <li>● erreur</li> <li>● réinitialisation du bloc</li> </ul> A l'état 0, une requête est en cours.
Sortie Défaut (Erreur)	%MSGx.E	A l'état 1, la communication est terminée si : <ul style="list-style-type: none"> <li>● commande incorrecte</li> <li>● table configurée de manière incorrecte</li> <li>● réception d'un caractère incorrect (vitesse, parité, etc.)</li> <li>● table de réception pleine (non mise à jour)</li> </ul> l'état 0, la longueur du message et la liaison sont correctes.

Si une erreur survient lors de l'exécution d'une instruction EXCH, les bits %MSGx.D et %MSGx.E sont mis à 1. Le mot système %SW63 contient le code de l'erreur du port 1 et le mot système %SW64 celui du port 2. Voir *Mots système (%SW)*, p. 555.

**Entrée RAZ (R)**

Lorsque l'entrée RAZ est mise à 1 :

- L'émission de tous les messages est interrompue.
- La sortie Défaut (Erreur) est remise à 0.
- Le bit Terminé est mis à 1.

Un nouveau message peut être envoyé.

**Sortie Défaut (Erreur) (%MSGx.E)**

La sortie Défaut est mise à 1 en cas d'erreur de programmation des communications ou d'erreur d'émission d'un message. La sortie Défaut est mise à 1 si le nombre d'octets définis dans le bloc de données associé à l'instruction EXCH (mot 1, octet de poids faible) est supérieur à 128 (+80 en hexadécimal par FA).

La sortie Défaut est également mise à 1 en cas de problème lors de l'envoi d'un message Modbus vers un périphérique Modbus. Dans ce cas, l'utilisateur devra vérifier la connexion et s'assurer que le périphérique de destination peut recevoir des communications Modbus.

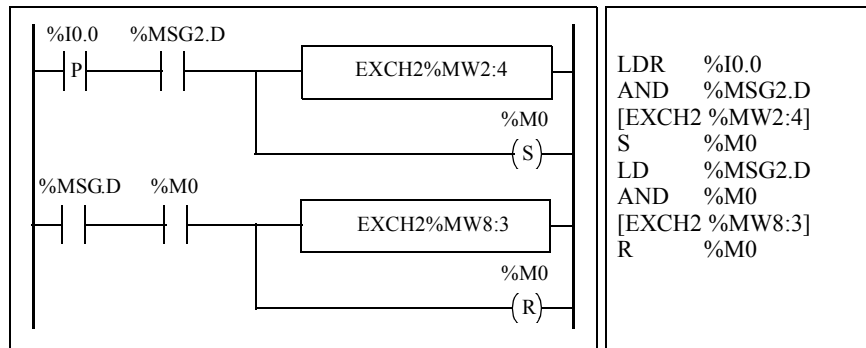
**Sortie Communication terminée (%MSGx.D)**

Lorsque la sortie Communication terminée est mise à 1, l'automate Twido est prêt à envoyer un autre message. L'utilisation de la sortie %MSGx.D est recommandée en cas d'envoi de plusieurs messages. Si cette sortie n'est pas utilisée, les messages pourront être perdus.

### Emission de plusieurs messages successifs

L'exécution de l'instruction EXCH permet d'activer un bloc message dans le programme d'application. Le message est émis si le bloc message n'est pas déjà actif (%MSGx.D = 1). Lorsque plusieurs messages sont envoyés au cours du même cycle, seul le premier message est émis. La gestion de l'émission de plusieurs messages à l'aide du programme incombe à l'utilisateur.

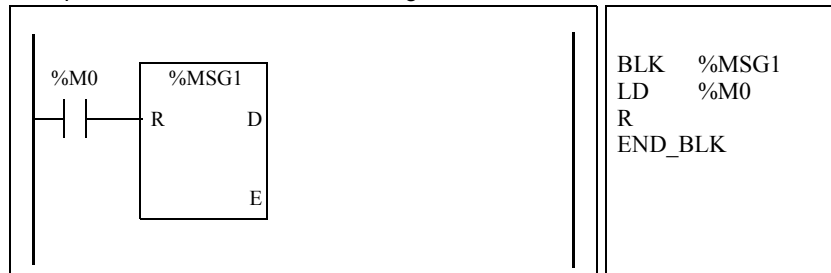
Exemple d'émission de deux messages successifs sur le port 2 :



### Réinitialisation des échanges

L'annulation d'un échange survient lors de l'activation de l'entrée (ou de l'instruction) R. Cette entrée initialise la communication, remet à 0 la sortie %MSGx.E et met la sortie %MSGx.D à 1. Notez qu'il est possible de réinitialiser une communication si une défaillance est détectée.

Exemple de réinitialisation d'un échange :



**Cas particuliers** Le tableau présente les cas particuliers de fonctionnement du bloc fonction %MSGx.

Cas particulier	Description
Effet d'un redémarrage à froid (%S0=1)	Force la réinitialisation de la communication.
Effet d'un redémarrage à chaud (%S1=1)	Aucun effet.
Effet d'un arrêt de l'automate	Si un message est en cours d'émission, l'automate interrompt le transfert et réinitialise les sorties %MSGx.D et %MSGx.E.

# 15.2 Fonctions horodateur

---

## Présentation

**Objet de ce sous-chapitre** Ce sous chapitre offre une description des fonctions de gestion du temps des automates Twido.

---

**Contenu de ce sous-chapitre** Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Fonctions horloges	437
Blocs horodateurs	438
Horodatage	441
Réglage de la date et de l'heure	443

---

---

## Fonctions horloges

---

### Introduction

Les automates Twido possèdent une fonction Date/Heure. Cette fonction requiert l'option Horodateur (RTC) et permet d'utiliser :

- Des **blocs horodateurs**, pour la programmation d'actions à des moments prédéfinis ou calculés.
- Une fonctionnalité d'**horodatage**, pour la consignation des durées et des calendriers d'événements et la mesure de la durée de ces derniers.

Pour accéder à l'horloge Date/Heure Twido, sélectionnez **Blocs horodateurs** dans le menu **Logiciel** de TwidoSoft. Notez que cette horloge peut également être réglée à l'aide d'un programme. En cas d'extinction de l'automate, les réglages de l'horloge sont conservés en mémoire pendant un maximum de 30 jours, si la batterie de l'automate était en charge pendant les six heures qui ont précédé l'extinction de l'automate.

L'affichage de l'horloge Date/Heure se fait au format « 24 heures » et tient compte des années bissextiles.

### Valeur de correction de l'horodateur

La définition de la valeur de correction de l'horodateur est nécessaire à son bon fonctionnement. Chaque horodateur possède sa propre valeur de correction, figurant au sein même de l'unité. Pour configurer cette valeur dans TwidoSoft, sélectionnez l'option **Configurer RTC** dans la boîte de dialogue **Actions automate**.

---

## Blocs horodateurs

### Introduction

Les blocs horodateurs permettent de programmer et de contrôler des actions selon un calendrier précis (mois, jour et heure). Un maximum de 16 blocs horodateurs peuvent être programmés. Ces blocs ne requièrent aucune saisie programme.

**Note :** Vérifiez le bit système %S51 et le mot système %SW118 afin de vous assurer que l'option horodateur (RTC) est installée. Reportez-vous à la rubrique *Bits système (%S)*, p. 546. L'option RTC est requise pour l'utilisation de blocs horodateurs.

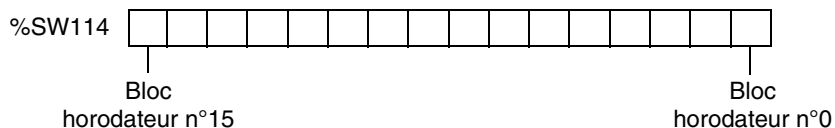
### Paramètres

Le tableau suivant répertorie les paramètres d'un bloc horodateur :

Paramètre	Format	Fonction/Plage
Numéro du bloc horodateur	n	n = 0 à 15
Configuré	Case à cocher	Cochez cette case pour configurer le bloc horodateur sélectionné.
Bit de sortie	%Qx.y.z	L'affectation de la sortie est activée par le bloc horodateur : %Mi ou %Qj.k. Cette sortie est mise à 1 lorsque les paramètres de date et d'heure courants sont compris entre les paramètres de début et de fin de la période active.
Mois de début	janvier à décembre	Mois au cours duquel débute le bloc horodateur.
Mois de fin	janvier à décembre	Mois au cours duquel s'achève le bloc horodateur.
Date de début	1 - 31	Jour au cours duquel débute le bloc horodateur.
Date de fin	1 - 31	Jour au cours duquel s'achève le bloc horodateur.
Heure de début	hh:mn	Heure à laquelle débute le bloc horodateur. Définie par l'heure (0 à 23), suivie des minutes (0 à 59).
Heure d'arrêt	hh:mn	Heure à laquelle s'achève le bloc horodateur. Définie par l'heure (0 à 23), suivie des minutes (0 à 59).
Jour de la semaine	lundi à dimanche	Cases à cocher permettant de définir les jours au cours desquels sera activé le bloc horodateur.

**Activation de blocs horodateurs**

Les bits du mot système %SW114 activent (lorsqu'ils sont mis à 1) ou désactivent (lorsqu'ils sont mis à 0) le fonctionnement des 16 blocs horodateurs. Affectation des blocs horodateurs dans %SW114 :



Par défaut (ou après un démarrage à froid), tous les bits de ce mot système sont mis à 1. L'utilisation de ces bits par le programme est optionnelle.

**Sortie des blocs horodateurs**

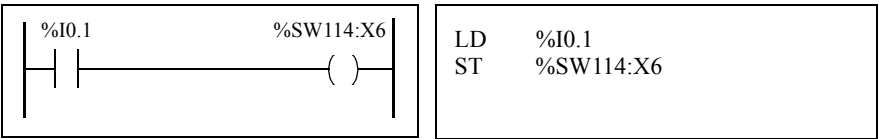
Si la même sortie (%Mi ou %Qj.k) est affectée par plusieurs blocs, c'est le OU des résultats de chacun des blocs qui est finalement affecté à cet objet (notez que la même sortie peut disposer de plusieurs « plages de fonctionnement »).

Exemple

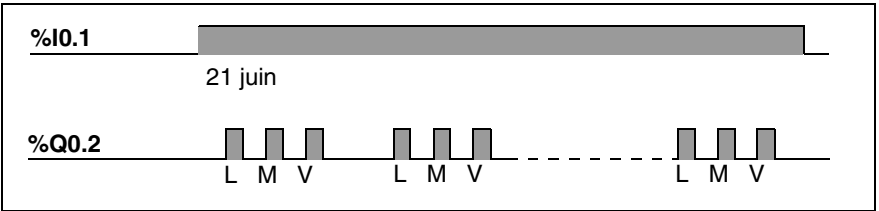
Le tableau suivant présente les paramètres d'un programme d'arrosage pendant la période d'été:

Paramètre	Valeur	Description
Bloc horodateur	6	Bloc horodateur numéro 6
Bit de sortie	%Q0.2	Activer la sortie %Q0.2
Mois de début	Juin	Débuter l'activité en juin
Mois de fin	Septembre	Arrêter l'activité en septembre
Date de début	21	Débuter l'activité le 21ème jour de juin
Date de fin	21	Arrêter l'activité le 21ème jour de septembre
Jour de la semaine	lundi, mercredi, vendredi	Exécuter l'activité les lundis, mercredis et vendredis
Heure de début	21:00	Débuter l'activité à 21:00
Heure d'arrêt	22:00	Arrêter l'activité à 22:00

Le programme suivant permet de désactiver le bloc horodateur grâce à un commutateur ou un détecteur d'humidité lié à l'entrée %I0.1.



Le chronogramme suivant illustre l'activation de la sortie %Q0.2.



Gestion de plage  
horaire par  
programme

Les paramètres de date et d'heure sont disponibles dans les mots système %SW50 à %SW53 (reportez-vous à la rubrique *Mots système (%SW)*, p. 555). Il est ainsi possible d'effectuer un horodatage dans le programme de l'automate en effectuant des comparaisons arithmétiques entre la date et l'heure courantes et les valeurs immédiates ou les mots %MWi (ou %KWi), qui peuvent contenir des consignes.



## Horodatage

### Introduction

Les mots système %SW49 à %SW53 contiennent les paramètres de date et d'heure au format BCD (reportez-vous à la section *Révision du code BCD*, p. 374), qui est utile pour l'affichage sur un périphérique ou la transmission vers ce périphérique. Ces mots système peuvent être utilisés pour stocker les paramètres de date et d'heure d'un événement (reportez-vous au sous-chapitre *Mots système (%SW)*, p. 555).

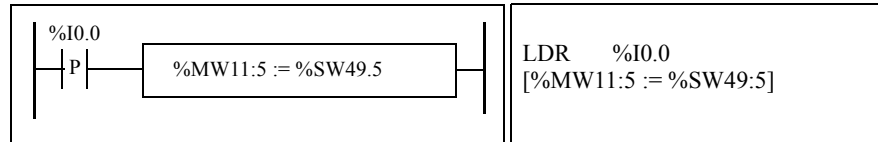
**Note :** Les paramètres de date et d'heure peuvent également être réglés à l'aide de l'afficheur optionnel (reportez-vous au sous-chapitre *Horloge calendaire*, p. 259).

### Datage d'un événement

Pour dater un événement, il suffit d'utiliser des opérations d'affectation, pour transférer le contenu de mots système vers des mots internes et de traiter ces mots internes (par exemple, la transmission vers l'afficheur à l'aide de l'instruction EXCH).

### Exemple de programmation

L'exemple suivant montre comment dater un front montant sur l'entrée %I0.1.



Dès qu'un événement est détecté, la table de mots contient :

Codage	Octet de poids fort	Octet de poids faible
%MW11		Jour de la semaine <sup>1</sup>
%MW12	00	Seconde
%MW13	Heure	Minute
%MW14	Mois	Jour
%MW15	Siècle	Année

**Note :** (1) 1 = lundi, 2 = mardi, 3 = mercredi, 4 = jeudi, 5 = vendredi, 6 = samedi, 7 = dimanche.

**Exemple de table  
de mots**

Exemple de données pour le lundi 19 avril 2002, à 13:40:30 :

Mot	Valeur (hexa.)	Signification
%MW11	0001	Lundi
%MW12	0030	30 secondes
%MW13	1340	13 heures, 40 minutes
%MW14	0419	04 = avril, le 19
%MW15	2002	2002

---

**Date et heure du  
dernier arrêt**

Les mots système %SW54 à %SW57 contiennent les paramètres de date et d'heure du dernier arrêt et le mot %SW58 contient le code affichant la cause du dernier arrêt, au format BCD (reportez-vous au sous-chapitre *Mots système (%SW)*, p. 555).

---

## Réglage de la date et de l'heure

---

### Introduction

Pour mettre à jour les paramètres de date et d'heure, vous pouvez utiliser l'une des méthodes suivantes :

- TwidoSoft

Utilisez la boîte de dialogue **Définir heure**. Celle-ci est accessible depuis la boîte de dialogue **Actions automate**. Pour afficher cette boîte de dialogue, sélectionnez **Actions automate** dans le menu **Automate**.

- Mots système

Utilisez les mots système %SW49 à %SW53 ou le mot système %SW59.

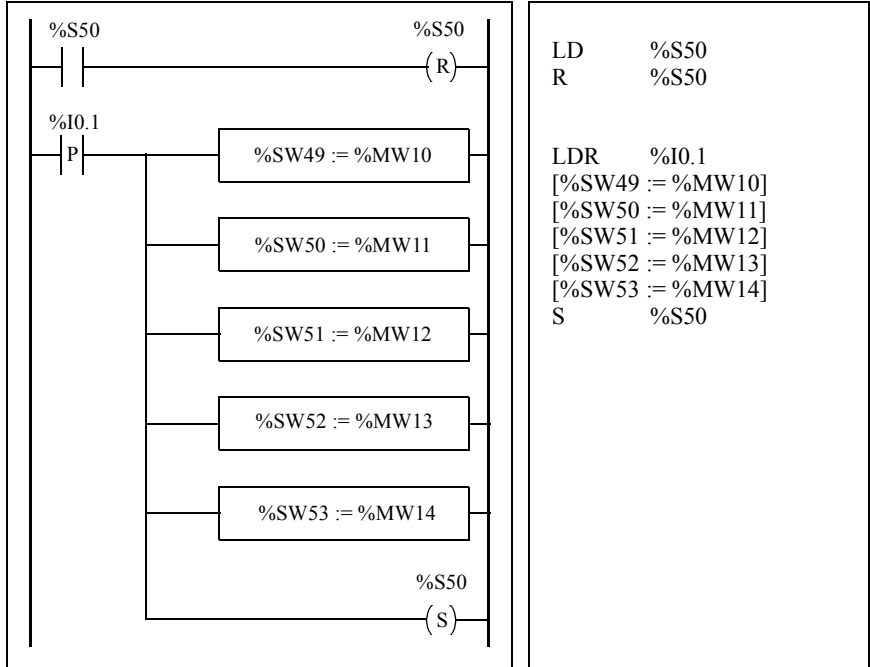
Les paramètres de date et d'heure peuvent être mis à jour uniquement lorsque la cartouche optionnelle de l'horodateur (TWDXCPRTC) est installée sur l'automate. Remarque : Les automates compacts TWDLCA•40DRF disposent d'un horodateur intégré.

---

Utilisation des mots %SW49 à %SW53

- Pour utiliser les mots système %SW49 à %SW53 afin de paramétrer la date et l'heure, le bit %S50 doit être mis à 1. Cela a pour conséquence :
- l'annulation de la mise à jour des mots %SW49 à %SW53 via l'horloge interne ;
  - le transfert des valeurs écrites dans les mots %SW49 à %SW53 vers l'horloge interne.

Exemple de programmation :



Les mots %MW10 à %MW14 contiendront les nouveaux paramètres de date et d'heure au format BCD (voir *Révision du code BCD*, p. 374) et correspondront au codage des mots %SW49 à %SW53.

La table de mots doit contenir les nouveaux paramètres de date et d'heure :

Codage	Octet de poids fort	Octet de poids faible
%MW10		Jour de la semaine <sup>1</sup>
%MW11		Seconde
%MW12	Heure	Minute
%MW13	Mois	Jour
%MW14	Siècle	Année

**Note :** (1) 1 = lundi, 2 = mardi, 3 = mercredi, 4 = jeudi, 5 = vendredi, 6 = samedi, 7 = dimanche.

Exemple pour le lundi 19 avril 2002 :

Mot	Valeur (hex.)	Signification
%MW10	0001	Lundi
%MW11	0030	30 secondes
%MW12	1340	13 heures, 40 minutes
%MW13	0419	04 = avril, le 19
%MW14	2002	2002

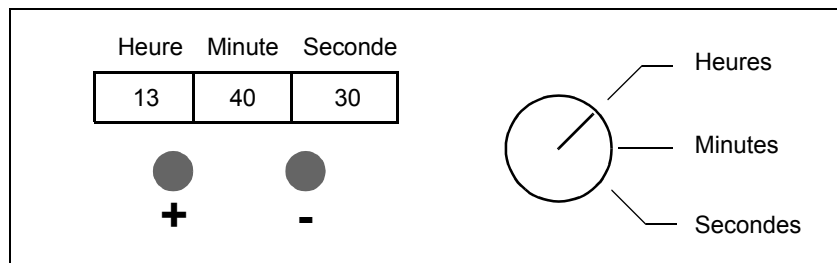
### Utilisation du mot %SW59

Pour mettre à jour la date et l'heure, vous pouvez également utiliser le bit système %S59 et le mot système %SW59 de réglage de la date.

La mise du bit %S59 à 1 permet de régler les paramètres de date et d'heure courants à l'aide du mot %SW59 (voir *Mots système (%SW)*, p. 555). Le mot système %SW59 permet d'incrémenter ou de décrémenter chacun des composants de date et d'heure sur un front montant.

**Exemple de mise en œuvre**

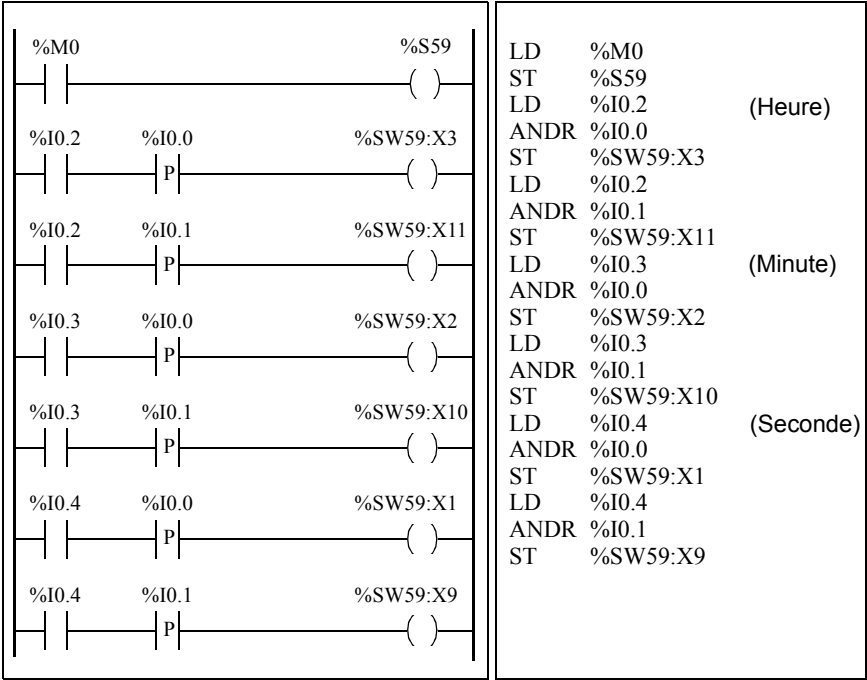
Le panneau avant ci-dessous permet de modifier le réglage de l'horloge interne (heures, minutes et secondes).



Description des commandes :

- Le commutateur Heures/Minutes/Secondes permet de sélectionner l'heure à modifier, respectivement à l'aide des entrées %I0.2, %I0.3 et %I0.4.
- Le bouton + permet d'incrémenter l'affichage de l'heure sélectionnée, à l'aide de l'entrée %I0.0.
- Le bouton - permet de décrémenter l'affichage de l'heure sélectionnée, à l'aide de l'entrée %I0.1.

Le programme suivant lit les entrées du panneau et règle l'horloge interne.



## 15.3            Fonction PID

---

### Présentation

---

**Objet de ce sous-chapitre**    Ce sous-chapitre décrit le comportement, les fonctionnalités et la mise en oeuvre de la fonction PID.

**Note :** Pour obtenir des informations de configuration sur l'automate PID, ainsi que sur l'auto tuning PID, consultez le **Guide de démarrage rapide du PID de l'automate Twido** disponible au format électronique sur le CD d'installation et de documentation TwidoSoft.

---



**Contenu de ce sous-chapitre**

Ce sous-chapitre contient les sujets suivants :

<b>Sujet</b>	<b>Page</b>
Présentation générale	450
Principe de la boucle de régulation	451
Méthodologie de développement d'une application de régulation	452
Compatibilités et performances	453
Caractéristiques détaillées de la fonction PID	454
Comment accéder à la configuration du PID	458
Onglet Général du PID	460
Onglet Entrée du PID	463
Onglet PID	466
Onglet Auto tuning de la fonction PID	469
Onglet Sortie du PID	474
Comment accéder à la mise au point du PID	477
Onglet Animation du PID	479
Onglet Trace du PID	482
Etats du PID et codes d'erreurs	485
Réglage PID avec la fonction d'auto tuning (AT)	489
Méthode de réglage des paramètres PID	500
Rôle et influence des paramètres d'un PID	504
Annexe 1 : Notions fondamentales de la théorie PID	508
Annexe 2 : Premier ordre avec modèle de temporisation	510

## Présentation générale

---

### Généralités

La fonction de régulation PID est une fonction du langage de programmation TwidoSoft.

Elle permet de programmer des boucles de régulation PID sur automates Twido de version supérieure ou égale à 1.2.

Cette fonction est particulièrement adaptée pour :

- répondre aux besoins de process séquentiels nécessitant des fonctions de régulation auxiliaire (exemples : machines d'emballage à film plastique, machines de traitement de surface, presses...),
- répondre aux besoins des process de régulation simple (exemples : fours de traitements de métaux, fours à céramiques, petits groupes frigorifiques...),

**Sa mise en oeuvre est très simple** car elle s'effectue par des écrans de :

- configuration,
- et de mise au point,

associés à une ligne de programme (bloc opération en langage à contact ou simple appel de PID en liste d'instruction) qui indique le numéro du PID utilisé.

Exemple de ligne de programme en langage à contact :



**Note :** dans une même application d'automatisme Twido le nombre maximum de fonctions PID configurables est de 14.

### Principales fonctionnalités

Les principales fonctionnalités sont les suivantes :

- entrée analogique,
  - conversion linéaire de la mesure configurable,
  - alarme haute et basse en entrée configurable,
  - sortie analogique ou PWM,
  - écrêtage de la sortie configurable,
  - action directe ou inverse configurable.
-

## Principe de la boucle de régulation

### Présentation

Le fonctionnement d'une boucle de régulation comprend trois phases distinctes :

- l'acquisition des données :
  - mesure(s) provenant des capteurs du process (analogiques, codeurs),
  - consigne(s) provenant généralement de variables internes de l'automate ou de données issues d'une table d'animation TwidoSoft.
- l'exécution de l'algorithme de régulation PID,
- l'envoi des commandes adaptées aux caractéristiques des actionneurs à piloter via des sorties TOR (PWM) ou analogiques.

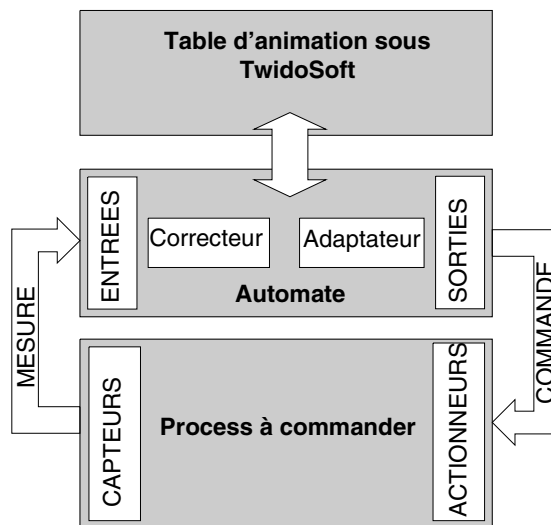
L'algorithme PID élabore le signal de commande à partir :

- de la mesure échantillonnée par le module d'entrée,
- de la valeur de la consigne fixée soit par l'opérateur, soit par programme,
- des valeurs des différents paramètres du correcteur.

Le signal issu du correcteur est soit traité directement par une carte de sortie analogique de l'automate raccordé à l'actionneur, soit traité via une adaptation PWM sur une sortie TOR de l'automate.

### Illustration

L'illustration ci-dessous schématise le principe d'une boucle de régulation.

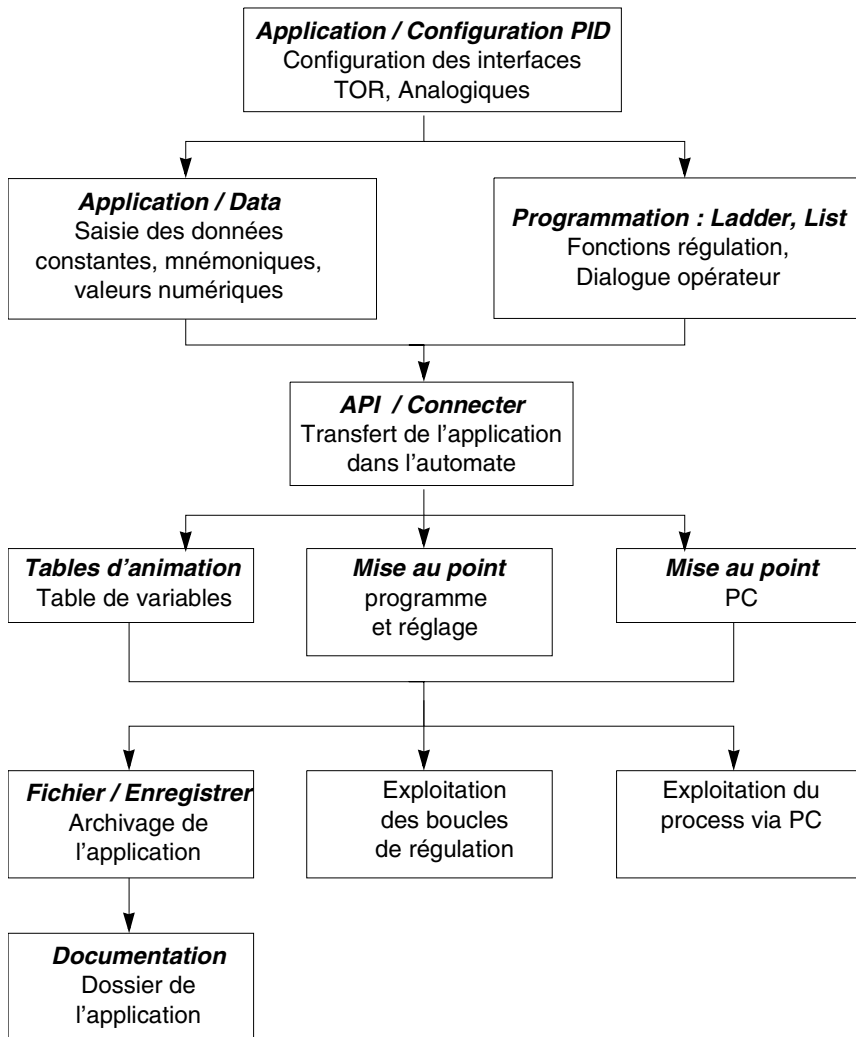


## Méthodologie de développement d'une application de régulation

### Schéma de principe

Le schéma ci-dessous présente l'ensemble des tâches à effectuer lors de la création et la mise au point d'une application de régulation.

**Note :** l'ordre défini dépend de votre propre méthode de travail, il est donné à titre indicatif.



---

## Compatibilités et performances

---

### Présentation

La fonction PID du Twido est une fonctionnalité disponible à partir de la version 1.2 du Twido c'est pourquoi sa mise en oeuvre est sujette à un certain nombre de compatibilités matérielles et logicielles décrites dans les paragraphes suivants. D'autre part cette fonctionnalité nécessite des ressources qui sont présentées dans le paragraphe **Performances**.

---

### Compatibilités

La fonction PID du Twido est disponible sur les Twido de version logicielle supérieure ou égale à 1.2.  
Si vous disposez de Twidos de version logicielle inférieure, vous pouvez mettre à jour le firmware afin de pouvoir utiliser cette fonction PID.

**Note** : les modules d'entrées et de sorties analogiques de version 1.0 sont utilisables en entrées ou sorties de PID sans nécessiter de mise à jour.

Pour pouvoir configurer et programmer un PID sur ces différentes versions de matériel vous devez posséder la version **1.2 du logiciel TwidoSoft**.

---

### Performances

Les boucles de régulation PID possèdent les performances suivantes :

Description	Durée
Temps d'exécution d'une boucle	0,4 ms

---

## Caractéristiques détaillées de la fonction PID

---

### Général

La fonction PID réalise une correction PID à partir d'une mesure et d'une consigne analogiques au format par défaut [0 – 10 000] et fournit une commande analogique au même format ou une modulation de largeur (PWM) sur une sortie TOR. Tous les paramètres PID sont décrits dans les fenêtres qui permettent de les configurer. Nous nous contentons ici de faire une synthèse des fonctions disponibles, d'indiquer leurs mesures et de décrire leur intégration à la fonction PID dans un synoptique de fonctionnement.

**Note :** Pour une utilisation en pleine échelle (résolution optimale), vous pouvez configurer votre entrée analogique connectée à la branche mesure du PID en 0-10 000, Toutefois, l'automate fonctionne correctement si vous utilisez la configuration par défaut (0-4 095).

**Note :** Pour que la régulation puisse fonctionner correctement l'automate Twido doit impérativement **être en mode périodique**. La fonction PID est alors exécutée périodiquement à chaque cycle et l'échantillonnage des données d'entrée PID respecte la période définie dans la configuration (voir tableau suivant).

---

## Détails des fonctions disponibles

Le tableau suivant indique les différentes fonctions disponibles et les échelles correspondantes :

Fonction	Echelle et commentaire
Conversion linéaire de l'entrée	Cette opération permet de convertir une valeur respectant le format 0 à 10 000 (résolution module d'entrée analogique) à une valeur comprise entre -32 768 et 32 767.
Gain proportionnel	Associé à un facteur de 100, sa valeur est comprise entre 1 et 10 000. Cela correspond à un gain variant de 0,01 à 100. <b>Remarque :</b> Si vous saisissez une valeur de gain incorrecte (gain négatif ou nul), TwidoSoft ignore ce réglage utilisateur et affecte automatiquement la valeur par défaut de 100 à ce facteur.
Temps d'intégrale	Associée à une base temps de 0,1 seconde, sa valeur est comprise entre 0 et 20 000. Cela correspond à un temps d'intégrale compris entre 0 et 2 000,0 secondes.
Temps de dérivée	Associée à une base temps de 0,1 seconde, sa valeur est comprise entre 0 et 10 000. Cela correspond à un temps de dérivé compris entre 0 et 1 000,0 secondes.
Période d'échantillonnage	Associée à une base temps de 0,01 seconde, sa valeur est comprise entre 1 et 10 000. Cela correspond à une période comprise entre 0,01 et 100 secondes.
Sortie PWM	Associée à une base temps de 0,1 seconde, sa valeur est comprise entre 1 et 500. Cela correspond à une période de modulation comprise entre 0,1 et 50 secondes.
Sortie analogique	Valeur comprise entre 0 et +10 000
Alarme haute sur la mesure	Cette alarme est définie après la conversion. Elle est définie sur une valeur comprise entre -32 768 et 32 767 si la conversion est activée et entre 0 et 10 000 dans le cas contraire.
Alarme basse sur la mesure	Cette alarme est définie après la conversion. Elle est définie sur une valeur comprise entre -32 768 et 32 767 si la conversion est activée, et entre 0 et 10 000 dans le cas contraire.

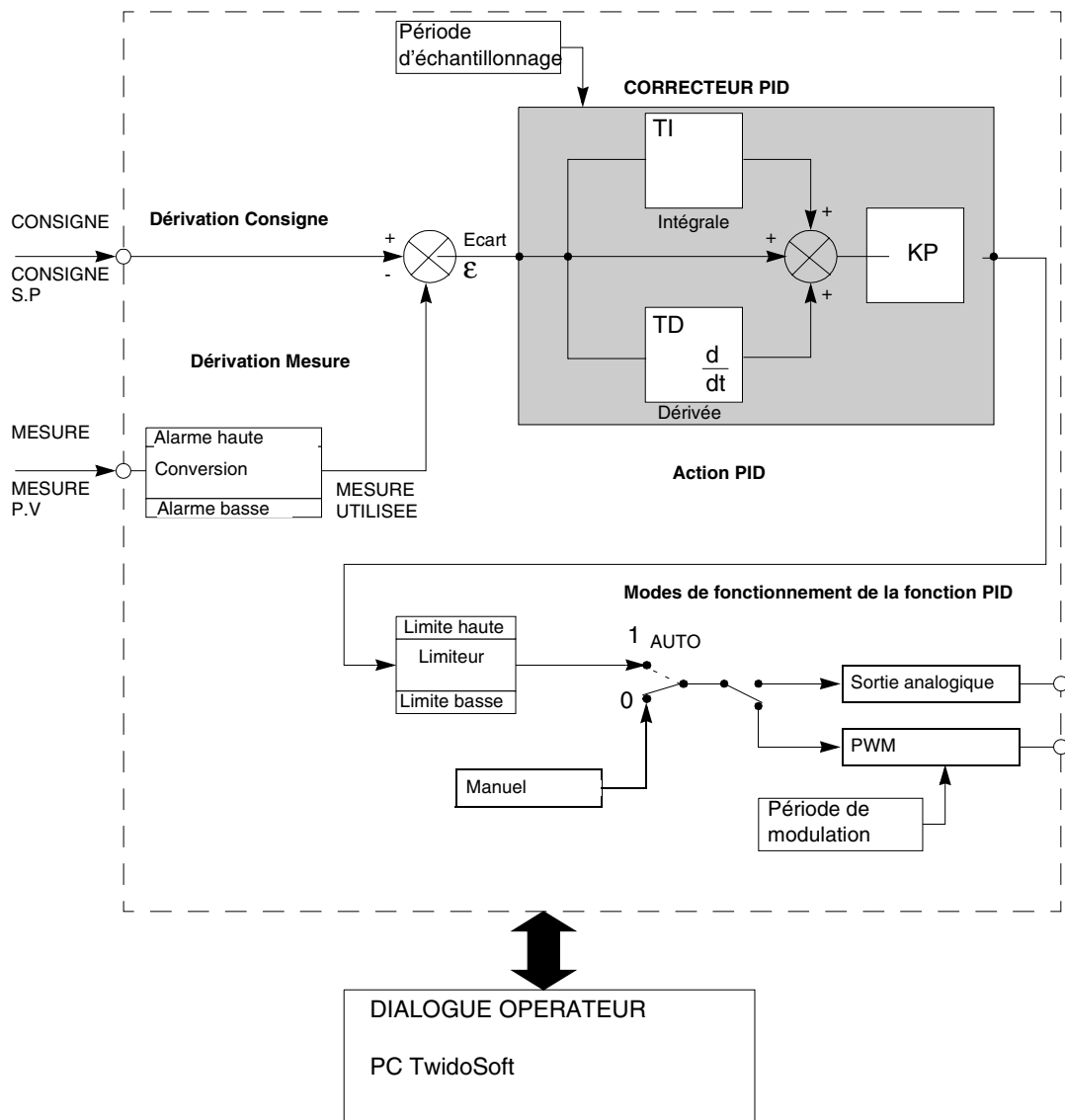
Fonction	Echelle et commentaire
Limite haute sur la sortie	Cette limite est comprise entre 0 et 10 000 pour une sortie analogique. Lorsque la fonction PWM est activée, la limite correspond à un pourcentage de la période modulée. 0% pour 0 et 100% pour 10 000.
Limite basse sur la sortie	Cette limite est comprise entre 0 et 10 000 pour une sortie analogique. Lorsque la fonction PWM est activée, la limite correspond à un pourcentage de la période modulée. 0 % pour 0 et 100 % pour 10 000.
Mode manuel	Lorsque le mode manuel est activé, la sortie est égale à une valeur fixe paramétrée par l'utilisateur. Cette sortie est comprise entre 0 et 10 000 (0 à 100 % pour sortie PWM).
Action directe ou inverse	L'action directe ou inverse est disponible et agit directement sur la sortie.
Auto tuning	Cette fonction permet de régler automatiquement les paramètres de Kp, Ti, Td et de l'action directe/inverse afin d'obtenir une convergence optimum de la régulation.

**Note :** Pour une meilleur compréhension de l'action de chacune des fonctions décrites dans le tableau précédent reportez-vous au synoptique qui suit.



## Principes de fonctionnement

Le schéma suivant présente le principe de fonctionnement de la fonction PID.

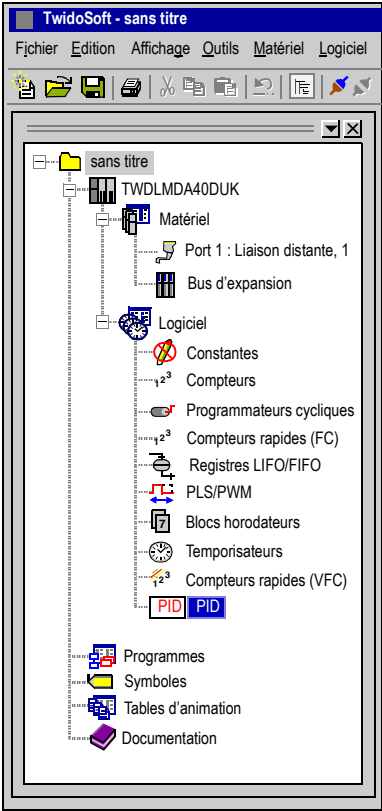


**Remarque :** La description des paramètres utilisés est présentée dans le tableau de la page précédente et dans les écrans de configuration.

## Comment accéder à la configuration du PID

**Présentation** L'accès aux écrans de configuration d'un PID sur automates TWIDO est décrit dans les paragraphes qui suivent.

**Marche à suivre** Le tableau suivant présente la marche à suivre pour accéder aux écrans de configuration d'un PID :

Etape	Action
1	Vérifiez que vous êtes en mode <b>local</b> .
2	Ouvrez le navigateur. <b>Résultat :</b> 

Etape	Action
3	<p>Double cliquez sur <b>PID</b>.</p> <p><b>Résultat</b> : la fenêtre de paramétrage des PID s'ouvre, elle est par défaut positionnée sur l'onglet <b>Général</b> (Voir <i>Onglet Général du PID</i>, p. 460).</p> <p><b>Note</b> : vous pouvez également effectuer un clic droit sur <b>PID</b> et choisir l'option <b>Editer</b> ou sélectionner le menu <b>Logiciel</b> → <b>PID</b> ou utiliser le menu <b>Programme</b> → <b>Editeur de configuration</b> → <b> Icône PID</b> ou bien, dans ce dernier cas, choisir le PID et cliquer sur l'icône <b>loupe</b> pour sélectionner un PID précis.</p>

## Onglet Général du PID

---

### Présentation

Lorsque vous ouvrez la fonction PID à partir du navigateur, la fenêtre de configuration correspondante apparaît. A partir de cette fenêtre, vous pouvez :

- configurer chaque PID de l'automate Twido ;
- mettre au point chaque PID de l'automate Twido.

Lorsque vous affichez cet écran et que vous êtes :

- en mode local : vous accédez à l'onglet par défaut **Général** et aux paramètres de configuration ;
- en mode connecté : vous accédez à l'onglet **Animation** et aux paramètres de mise au point et de réglage.

**Note :** Dans certains cas, les onglets et les champs grisés ne sont pas accessibles pour une des deux raisons suivantes : Le mode "PID uniquement" est sélectionné ce qui interdit l'accès aux paramètres de l'onglet AT qui ne sont plus nécessaires.

- Le mode de fonctionnement (local ou connecté) qui est actuellement activé ne permet pas d'accéder à ces paramètres.
- Le mode "PID uniquement" est sélectionné, interdisant ainsi l'accès aux paramètres de l'onglet Auto tuning qui ne sont plus nécessaires.

Les paragraphes qui suivent décrivent l'onglet **Général**.

---

## Onglet Général de la fonction PID

L'écran suivant permet de renseigner les paramètres généraux du PID.

The screenshot shows a software window titled "PID" with a standard Windows-style title bar (minimize, maximize, close buttons). Below the title bar is a text field labeled "Numéro PID" containing the value "0".

The main area of the window is divided into tabs: "Généralités", "Entrée", "PID", "AT", "Sortie", "Animation", and "Trace". The "Généralités" tab is currently selected.

Under the "Généralités" tab, there are several configuration options:

- "Mode de fonctionnement" is set to "PID" via a dropdown menu.
- A checked checkbox labeled "Configuré" is present.
- A text field labeled "Adresse mot" is empty.
- A checked checkbox labeled "Etats du PID" is followed by an empty text field.

Below these options is a functional block diagram:

- The "PID" section contains a "Consigne" (Setpoint) block.
- The "Entrée" (Input) section contains a "Mes" (Measurement) block with a graph icon and a warning triangle icon.
- The "AT" (Anti-windup) section contains a "PV Limite" (Process Variable Limit) block.
- The "Automate PID" (PID Controller) block is the central component.
- The "Sortie" (Output) section contains a "D/I" (Derivative/Integral) block, a graph icon, a hand icon, and a pulse icon.

Connections in the diagram:

- "Consigne" is connected to a summing junction (+/-) which feeds into the "Automate PID" block.
- "Mes" is connected to the summing junction.
- "PV Limite" is connected to the "Automate PID" block.
- The "Automate PID" block is connected to the "D/I" block.
- The "D/I" block is connected to the "Sortie" section.
- A thick black arrow points from the "AT" block up to the "Automate PID" block.

At the bottom of the window are five buttons: "OK", "Annuler", "Précédent", "Suivant", and "Aide". The "OK" button is highlighted with a black border.

**Description**

Le tableau suivant décrit les paramètres que vous pouvez définir.

Champ	Description
<b>Numéro PID</b>	Indiquez ici le numéro de la fonction PID à configurer. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
<b>Configuré</b>	Cette case doit être cochée pour pouvoir configurer la fonction PID. Si tel n'est pas le cas, aucune action ne peut être effectuée dans ces écrans et la fonction PID, bien qu'existant dans l'application, ne peut pas être utilisée.
<b>Mode de marche</b>	Indiquez ici le mode de fonctionnement désiré. Vous pouvez choisir entre trois modes de fonctionnement et une adresse de mot, comme suit : <ul style="list-style-type: none"><li>● PID</li><li>● AT</li><li>● PID+Auto tuning</li><li>● Adresse Mot</li></ul>
<b>Adresse Mot</b>	Vous pouvez définir un mot interne (%MW0 à %MW2999) dans la zone de texte. Ce mot est utilisé pour définir le mode de fonctionnement par programme. Le mot interne accepte trois valeurs possibles selon le mode de fonctionnement que vous souhaitez définir : <ul style="list-style-type: none"><li>● %MWx = 1 (pour définir PID uniquement)</li><li>● %MWx = 2 (pour définir PID+Auto tuning)</li><li>● %MWx = 3 (pour définir Auto tuning uniquement)</li></ul>
<b>Etats du PID</b>	Si vous cochez cette option, vous pouvez définir un mot mémoire dans cette zone de texte (%MW0 à %MW2999). Ce mot est utilisé par l'automate PID pour enregistrer l'état PID courant lors de l'exécution de l'automate PID et/ou la fonction d'auto tuning (pour plus de détails, reportez-vous à la section <i>Etats du PID et codes d'erreurs</i> , p. 485.)
<b>Schéma</b>	Le schéma vous permet de visualiser les différentes configurations possibles de la fonction PID.

## Onglet Entrée du PID

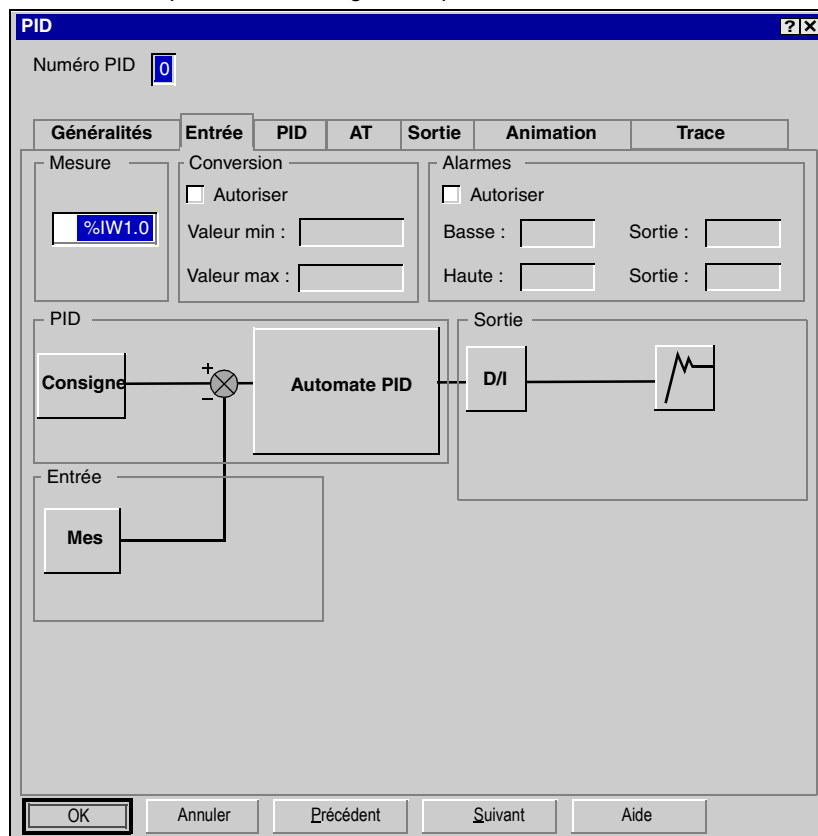
### Présentation

Cet onglet permet de renseigner les paramètres d'entrée du PID.

**Note** : Il est accessible en mode local.

### Onglet Entrée de la fonction PID

L'écran suivant permet de renseigner les paramètres d'entrée du PID.



**Description**

Le tableau suivant décrit les paramètres que vous pouvez définir.

Champ	Description
<b>Numéro PID</b>	Indiquez ici le numéro de la fonction PID à configurer. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
<b>Mesure</b>	Indiquez ici la variable qui contiendra la valeur de la mesure du processus à réguler. L'échelle par défaut est 0 à 10 000. Vous pouvez saisir soit un mot interne (%MW0 à %MW2999) soit une entrée analogique (%IWx.0 à %IWx.1).
<b>Conversion</b>	Cochez cette case si vous désirez effectuer une conversion de la variable du processus indiquée en entrée du PID. Si cette case est cochée, les deux champs <b>Valeur min</b> et <b>Valeur max</b> sont accessibles. La conversion est linéaire et convertit une valeur comprise entre 0 et 10 000 en une valeur dont les minimums et maximums peuvent être compris entre -32 768 et +32 767.
<b>Valeur min</b> <b>Valeur max</b>	Indiquez les valeurs minimum et valeurs maximum de l'échelle de conversion. La variable du processus est ensuite réévaluée automatiquement dans l'intervalle <b>[Valeur min à valeur max]</b> . <b>Remarque</b> : la <b>Valeur min</b> doit obligatoirement être inférieur à la <b>Valeur max</b> . <b>Valeur min</b> ou <b>Valeur max</b> peuvent être soit des mots internes (%MW0 à %MW2999) soit des constantes internes (%KW0 à %KW255) soit une valeur comprise entre -32 768 et +32 767.
<b>Alarmes</b>	Cochez cette case si vous désirez activer des alarmes sur des variables d'entrée. <b>Remarque</b> : les valeurs d'alarme sont à déterminer par rapport à la variable obtenue après la phase de conversion. Elles doivent par conséquent être comprises entre <b>Valeur min</b> et <b>Valeur max</b> lorsque la conversion est activée, sinon elles seront compris entre <b>0 et 10 000</b> .
<b>Basse</b> <b>Sortie</b>	Indiquez la valeur d'alarme haute dans le champ <b>Basse</b> . Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. <b>Sortie</b> doit contenir l'adresse du bit qui sera mis à 1 lorsque la limite basse est atteinte. <b>Sortie</b> peut être soit un bit interne (%M0 à %M255) soit une sortie (%Qx.0 à %Qx.32).



---

Champ	Description
<b>Haute Sortie</b>	Indiquez la valeur d'alarme basse dans le champs <b>Haute</b> . Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. <b>Sortie</b> doit contenir l'adresse du bit qui sera mis à 1 lorsque la limite haute est atteinte. <b>Sortie</b> peut être soit un bit interne (%M0 à %M255) soit une sortie (%Qx.0 à %Qx.32).
<b>Synoptique</b>	Le schéma vous permet de visualiser les différentes configurations possibles de la fonction PID.

---

## Onglet PID

### Présentation

Cet onglet permet de renseigner les paramètres internes de la fonction PID.

**Note :** Il est accessible en mode local.

### Onglet PID de la fonction PID

L'écran suivant permet de renseigner les paramètres internes du PID.

The screenshot displays a software interface for configuring a PID controller. At the top, a blue title bar reads "PID" with a help icon. Below it, a "Numéro PID" field contains the value "0". A series of tabs are visible: "Généralités", "Entrée", "PID" (which is selected), "AT", "Sortie", "Animation", and "Trace".

The "PID" tab is divided into three main sections:

- Consigne:** A single input field.
- Paramètres:** Three input fields for "Kp (x 0.01)", "Ti (0,1 s)", and "Td (0,1 s)".
- Période d'échantillonnage:** A label "(10 ms)" above a numeric input field set to "500".

Below these sections is a functional block diagram:

- The "Consigne" input is connected to a summing junction (a circle with a cross).
- The "Entrée" section contains a "Mes" (Measurement) input, which is also connected to the same summing junction.
- The output of the summing junction is fed into a block labeled "Automate PID".
- The output of the "Automate PID" block goes to a "D/I" (Derivative/Integral) block.
- The final output of the "D/I" block is shown as a waveform on a graph.

At the bottom of the window, there are five buttons: "OK", "Annuler", "Précédent", "Suivant", and "Aide". The "OK" button is highlighted with a black border.

**Description**

Le tableau suivant décrit les paramètres que vous pouvez définir.

Champ	Description
<b>Numéro PID</b>	Indiquez ici le numéro de la fonction PID à configurer. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
<b>Consigne</b>	Saisissez ici la valeur de consigne de la fonction PID. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. Elle doit donc être comprise entre 0 et 10 000 lorsque la conversion est inhibée. Dans les autres cas, la valeur doit être comprise en la <b>valeur minimale</b> et la <b>valeur maximale</b> pour la conversion.
<b>Kp * 100</b>	Indiquez ici le coefficient proportionnel de la fonction PID multiplié par 100. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. La plage valide pour la paramètre Kp est : $0 < Kp < 10\,000$ . <b>Remarque</b> : Si le paramètre Kp est défini par erreur sur 0 ( $Kp \leq 0$ est invalide), la valeur par défaut $Kp=100$ est automatiquement affectée par la fonction PID.
<b>Ti (0,1 s)</b>	Indiquez ici le coefficient d'action intégrale par rapport à une base temps de 0,1 seconde. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. Elle doit être comprise entre 0 et 20 000. <b>Remarque</b> : Pour désactiver l'action intégrale du PID, réglez ce coefficient sur 0.
<b>Td (0,1 s)</b>	Indiquez ici le coefficient d'action dérivée par rapport à une base temps de 0,1 seconde. Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. Elle doit être comprise entre 0 et 10 000. <b>Remarque</b> : Pour désactiver l'action dérivée du PID, réglez ce coefficient sur 0.
<b>Période d'échantillonnage</b>	Indiquez ici la période d'échantillonnage du PID par rapport à une base de temps de $10^{-2}$ secondes (10 ms). Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe. Elle doit être comprise entre 1 (0,01 s) et 10 000 (100 s).
<b>Schéma</b>	Le schéma vous permet de visualiser les différentes configurations possibles de la fonction PID.

**Note** : Lorsque la fonction d'auto tuning est activée, les paramètres Kp, Ti et Td ne sont plus définis par l'utilisateur, car ils sont définis automatiquement et par programme par l'algorithme d'auto tuning. Dans ce cas, vous devez saisir dans ces champs un **mot interne** uniquement (%MW0 à %MW2999).

Attention : Ne saisissez pas de constante interne ou de valeur directe lorsque la fonction d'auto tuning est activée, car cela déclencherait une erreur lors de l'exécution de la fonction PID.

---

## Onglet Auto tuning de la fonction PID

---

### Présentation

La configuration des paramètres de la fonction PID peut se révéler une tâche ardue, laborieuse et propice à la création d'erreurs. Toutes ces raisons rendent la régulation difficile à réaliser pour les personnes expérimentées, mais pas nécessairement expertes en matière de régulation. Il est ainsi parfois difficile d'effectuer un réglage parfait.

L'algorithme d'auto tuning de la fonction PID permet de déterminer automatiquement et correctement les quatre éléments du PID suivants :

- le facteur de gain,
- la valeur de l'intégrale,
- la valeur de la dérivée,
- et l'action directe ou inverse.

La fonction d'auto tuning permet ainsi de régler rapidement et efficacement la boucle du processus.

---

### Exigences relatives à la fonction d'auto tuning

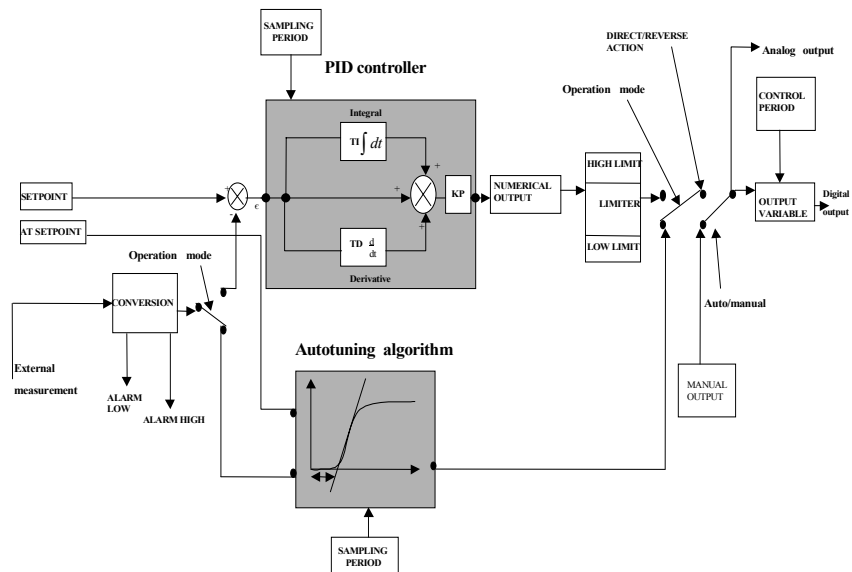
L'auto tuning de la fonction PID est particulièrement adapté à la régulation de température.

D'une manière générale, les processus utilisés par la fonction d'auto tuning pour réguler la température doivent répondre aux exigences suivantes :

- la régulation doit être principalement linéaire sur toute la plage de fonctionnement ;
  - la réponse de la régulation à une modification de niveau de sortie analogique doit suivre un schéma transitoire asymptotique ;
  - très peu de perturbations doivent se produire au sein des variables de régulation. (Dans le cas d'une régulation de température, vous ne devez pas constater de taux anormalement élevés d'échange de température entre la régulation et son environnement.)
-

## Principe de fonctionnement de l'auto tuning

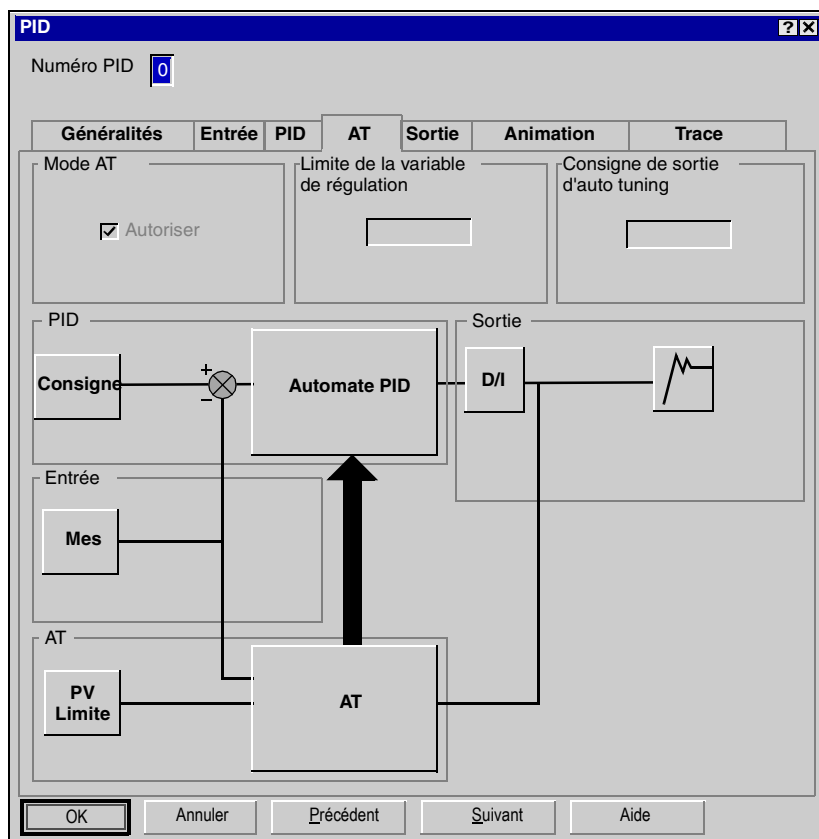
Le schéma suivant décrit le principe de fonctionnement de la fonction Auto tuning, ainsi que son interaction avec les boucles PID.




## Onglet Auto tuning de la fonction PID

L'écran suivant permet d'activer/désactiver la fonction d'auto tuning et de paramétrer cette dernière.

**Note** : Il est accessible en mode local uniquement.



## Description

	<b>AVERTISSEMENT</b>
	<p><b>La limite de la variable de régulation et les valeurs de consigne de sortie doivent être définies avec attention.</b></p> <p>L'auto tuning de la fonction PID est un processus de boucle ouverte qui agit directement sur le processus de contrôle sans régulation ni autre limitation que celles définies par la limite de la mesure et la consigne de sortie. Vous devez donc sélectionner soigneusement les deux valeurs avec des valeurs comprises dans la plage autorisée, conformément au processus, et ce afin d'éviter toute éventuelle surcharge.</p> <p><b>Le non-respect de ces précautions peut entraîner la mort, des lésions corporelles graves ou des dommages matériels.</b></p>

Le tableau suivant décrit les paramètres que vous pouvez définir.

Champ	Description
Autoriser	<p>Cochez cette case si vous souhaitez activer le mode Auto tuning. Vous pouvez utiliser la case à cocher de deux façons, suivant que vous définissez le mode de fonctionnement manuellement ou via une adresse mot dans l'onglet Général de la fonction PID.</p> <ul style="list-style-type: none"> <li>● Si vous définissez le <b>Mode de fonctionnement</b> sur <b>PID+Auto tuning</b> ou sur <b>Auto tuning</b> dans l'onglet <b>Général</b> (voir <i>Onglet Général du PID</i>, p. 460), l'option Autoriser est automatiquement cochée et grisée (vous ne pouvez pas la décocher).</li> <li>● Si vous définissez le mode de fonctionnement via une adresse mot %MWx (%MWx = 2: PID+AT; %MWx = 3: AT), vous devez cocher l'option Autoriser manuellement pour autoriser la configuration des paramètres d'auto tuning.</li> </ul> <p><b>Résultat :</b> Dans ces deux cas, tous les champs de l'écran de configuration de l'onglet Auto tuning sont activés et vous devez saisir les valeurs adéquates dans les champs de la consigne et de la sortie.</p>
Limite de la variable de régulation	<p>Spécifiez la limite de la variable de régulation au cours du processus d'auto tuning. Ce paramètre offre une sécurité au système de contrôle, l'auto tuning étant un processus boucle ouverte.</p> <p>Cette valeur peut être un mot interne (%MW0 à un maximum de %MW2999, selon la quantité de mémoire système disponible), une constante interne (%KW0 à %KW255) ou une valeur directe.</p> <p>Cette valeur doit donc être comprise entre 0 et 10 000 lorsque la conversion est inhibée. Dans les autres cas, la valeur doit être comprise entre les valeurs minimum et maximum pour la conversion.</p>



Champ	Description
Consigne de sortie d'auto tuning	<p>Saisissez ici la valeur de la sortie d'auto tuning. Il s'agit de la valeur du changement d'étape appliquée au processus.</p> <p>Cette valeur peut être un mot interne (%MW0 à %MW2999), une constante interne (%KW0 à %KW255) ou une valeur directe.</p> <p>La valeur doit donc être comprise entre 0 et 10 000.</p> <p><b>Remarque</b> : La consigne de sortie d'auto tuning doit toujours être supérieure à la dernière sortie appliquée au processus.</p>

**Note** : Lorsque la fonction d'auto tuning est activée, les constantes (%KWx) ou les valeurs directes ne sont plus autorisées. Seuls les mots mémoire sont autorisés dans les champs PID suivants :

- Les paramètres **Kp**, **Ti** et **Td** doivent être définis en tant que **mots mémoire** (%MWx) dans l'onglet PID.
- Le champ **Action** est automatiquement défini sur **Bit adresse** dans l'onglet Sortie de la fonction PID.
- La case **Bit** doit être complétée par un **bit mémoire** (%Mx) approprié dans l'onglet OUT.

### Coefficients calculés Kp, Ti et Td

Lorsque le processus d'auto tuning est terminé, les coefficients PID calculés Kp, Ti et Td :

- sont stockés dans leurs mots mémoire (%MWx) respectifs ;
- et apparaissent dans l'onglet **Animation**, en mode TwidoSoft connecté uniquement.

## Onglet Sortie du PID

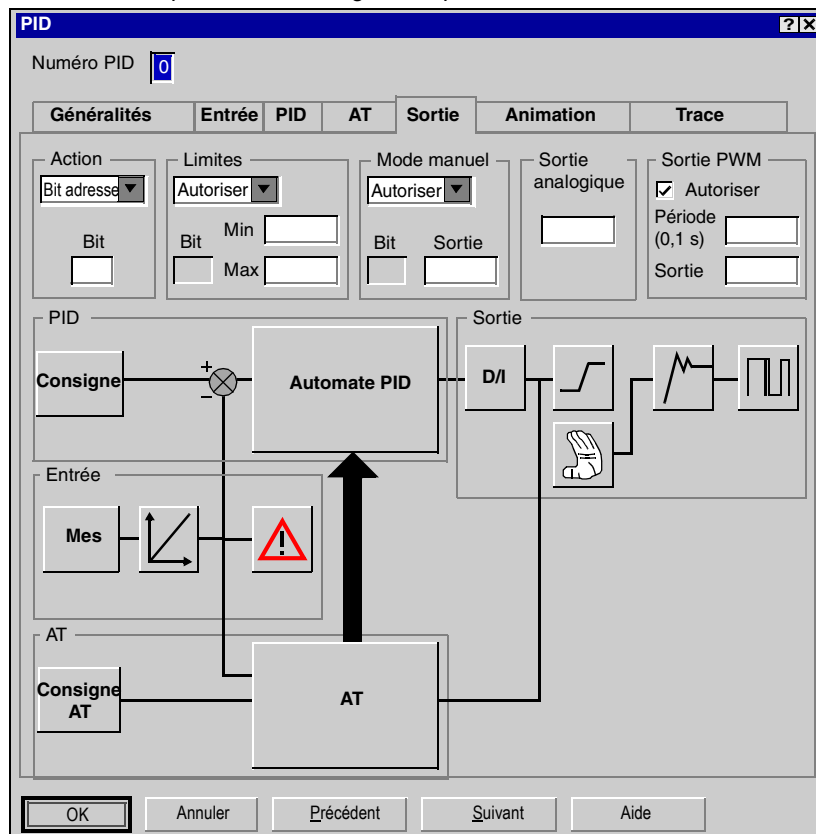
### Présentation

Cet onglet permet de renseigner les paramètres de sortie de la fonction PID.

**Note :** Il est accessible en mode local.

### Onglet Sortie de la fonction PID

L'écran suivant permet de renseigner les paramètres internes du PID.



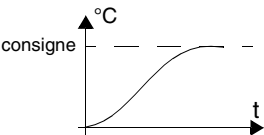
**Description**

Le tableau suivant décrit les paramètres que vous pouvez définir.

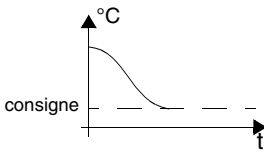
Champ	Description
<b>Numéro PID</b>	Indiquez ici le numéro de la fonction PID à configurer. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
<b>Action</b>	Indiquez ici le type d'action de la fonction PID sur le processus. Trois options sont disponibles : <b>Inverse</b> , <b>Directe</b> ou <b>adresse bit</b> . Si vous avez sélectionné <b>adresse bit</b> , vous pouvez modifier ce type par programme, en modifiant le bit associé qui est soit un bit interne (%M0 à %M255), soit une entrée (%Ix.0 à %Ix.32). L'action est directe si le bit est à 1, et inverse dans le cas contraire. <b>Remarque</b> : Lorsque la fonction AT est activée, l'algorithme d'auto tuning détermine automatiquement le type d'action approprié (directe ou inverse) pour le processus de contrôle. Dans ce cas, une seule option est disponible dans la liste déroulante Action : <b>Bit adresse</b> . Vous devez ensuite saisir dans la zone de texte du <b>Bit</b> associé, un <b>mot interne</b> (%MW0 à %MW2999). Ne tentez pas de saisir une constante interne ou une valeur directe dans la zone de texte <b>Bit</b> , car cela déclenche une erreur d'exécution.
<b>Limites Bit</b>	Indiquez ici si vous désirez limiter la sortie de la fonction PID. Trois options sont disponibles : <b>Autoriser</b> , <b>Inhiber</b> ou <b>adresse bit</b> . Si vous avez sélectionné <b>adresse bit</b> , vous pouvez autoriser (bit à 1) ou inhiber (bit à 0) la gestion des limites par programme, en modifiant le bit associé qui est soit un bit interne (%M0 à %M255) soit une entrée (%Ix.0 à %Ix.32).
<b>Min. Max.</b>	Définissez ici les limites haute et basse pour la sortie du PID. <b>Remarque</b> : La valeur <b>Mini</b> doit obligatoirement être inférieure à la valeur <b>Maxi</b> . <b>Min.</b> ou <b>Max.</b> peuvent être soit des mots internes (%MW0 à %MW2999), soit des constantes internes (%KW0 à %KW255), soit une valeur comprise entre 1 et 10 000.
<b>Mode manuel Bit Sortie</b>	Indiquez ici si vous désirez activer le mode manuel pour la fonction PID. Trois options sont disponibles : <b>Autoriser</b> , <b>Inhiber</b> ou <b>adresse bit</b> . Si vous avez sélectionné <b>adresse bit</b> , vous pouvez passer en mode manuel (bit à 1) ou passer en mode automatique (bit à 0) par programme, en modifiant le bit associé qui est soit un bit interne (%M0 à %M255) soit une entrée (%Ix.0 à %Ix.32). La <b>Sortie</b> du mode manuel doit contenir la valeur que vous désirez affecter à la sortie analogique lorsque le PID est en mode manuel. Cette <b>Sortie</b> peut être soit un mot (%MW0 à %MW2999) soit une valeur directe au format [0-10 000].

Champ	Description
Sortie analogique	Indiquez ici la sortie de la fonction PID en mode automatique. Cette <b>Sortie analogique</b> peut être de type %MW (%MW0 à %MW2999) ou %QW (%QWx.0).
Sortie PWM activée Période (0,1 s) Sortie	Cochez la case si vous souhaitez utiliser la fonction PWM de PID. Spécifiez la période de modulation dans <b>Période (0,1 s)</b> . Cette période doit être comprise entre 1 et 500. Elle peut être un mot interne (%MW0 à %MW2999) ou une constante interne (%KW0 à %KW255). Indiquez dans la valeur <b>Sortie</b> le bit de sortie PWM. Il peut s'agir d'un bit interne (%M0 à %M255) ou d'une sortie (%Qx.0 à %Qx.32).
Synoptique	Le synoptique vous permet de visualiser les différentes configurations possibles du PID.

**Note :** Le terme Inverse dans le champ action est utilisé pour atteindre une consigne haute (ex. : pour chauffer)



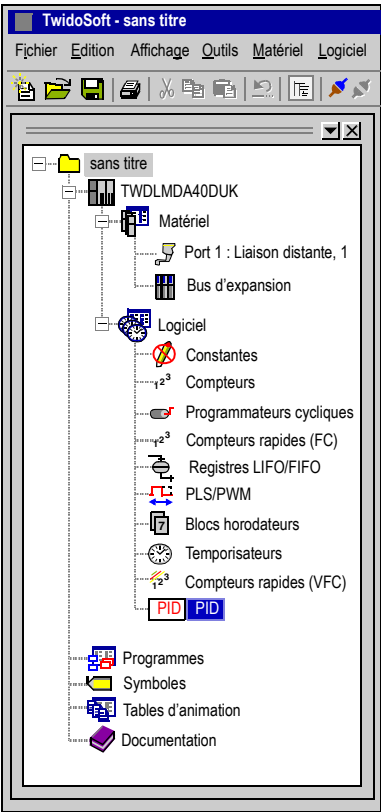
Le terme Directe dans le champ Action est utilisé pour atteindre une consigne basse (ex. : pour refroidir)



## Comment accéder à la mise au point du PID

**Présentation** L'accès aux écrans de mise au point d'un PID sur automates TWIDO est décrit dans les paragraphes qui suivent.

**Marche à suivre** Le tableau suivant présente la marche à suivre pour accéder aux écrans de mise au point d'un PID :

Etape	Action
1	Vérifiez que vous êtes en mode <b>connecté</b> .
2	Ouvrez le navigateur. <b>Résultat :</b> 

Etape	Action
3	<p>Double cliquez sur <b>PID</b>.</p> <p><b>Résultat</b> : la fenêtre de paramétrage des PID s'ouvre, elle est par défaut positionnée sur l'onglet <b>Animation</b> (Voir <i>Onglet Animation du PID</i>, p. 479).</p> <p><b>Note</b> : vous pouvez également effectuer un clic droit sur <b>PID</b> et choisir l'option <b>Editer</b> ou sélectionner le menu <b>Logiciel</b> → <b>PID</b> ou utiliser le menu <b>Programme</b> → <b>Editeur de configuration</b> → <b>Icône PID</b> ou bien, dans ce dernier cas, choisir le PID et cliquer sur l'icône <b>loupe</b> pour sélectionner un PID précis.</p>

---

## Onglet Animation du PID

---

### Présentation

Cet onglet permet d'effectuer la mise au point du PID.

Le synoptique affiché dépend du type de PID que vous avez créé, seuls les éléments configurés apparaissent.

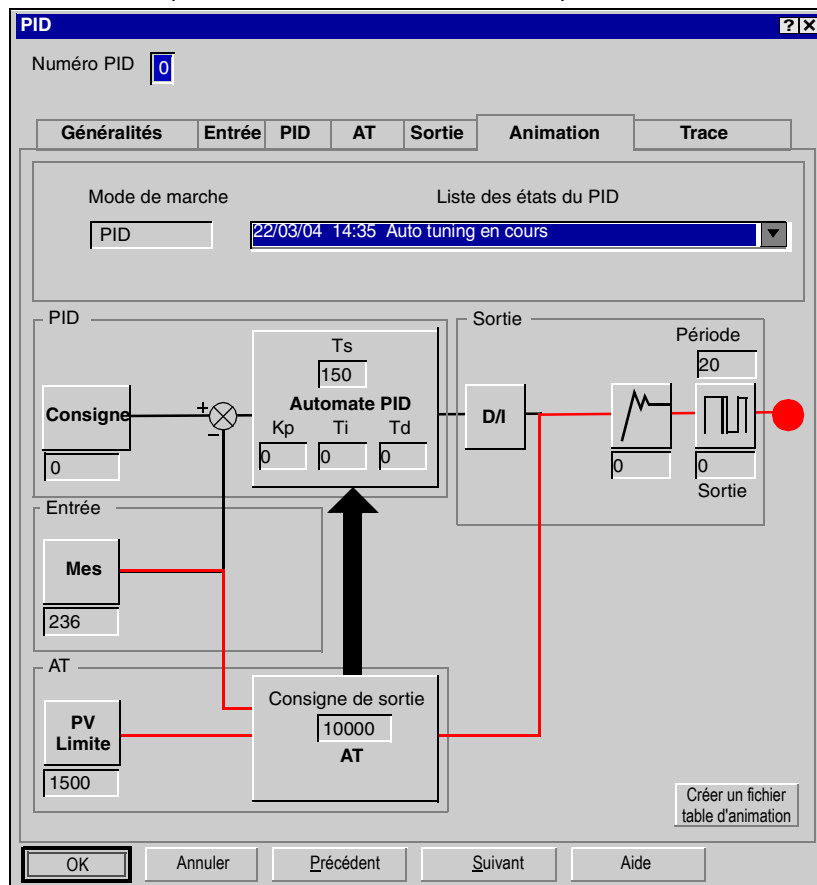
La visualisation est dynamique, les liaisons actives apparaissent en rouge, les liaisons inactives apparaissent en noir.

<b>Note</b> : Il est accessible en mode connecté.
---

---

## Onglet Animation du PID

L'écran suivant permet de visualiser et de mettre au point le PID.





**Description**

Le tableau suivant décrit les différentes zones de la fenêtre.

Champ	Description
<b>Numéro PID</b>	Indiquez ici le numéro du PID que vous désirez mettre au point. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
<b>Mode de marche</b>	Ce champ affiche le mode de marche courant du PID.
<b>Liste des états du PID</b>	Cette liste déroulante permet de visualiser en temps réel les 15 derniers états du PID. Chaque changement d'état met à jour cette liste en indiquant la date et l'heure ainsi que l'état courant.
<b>Créer un fichier table d'animation</b>	Cliquez sur le bouton <b>Créer un fichier table d'animation</b> , pour créer un fichier contenant toutes les variables visualisées sur le synoptique afin de vous permettre de les modifier en ligne et d'effectuer la mise au point de votre PID.

## Onglet Trace du PID

---

### Présentation

Cet onglet permet de visualiser le fonctionnement du PID et d'effectuer ses réglages.

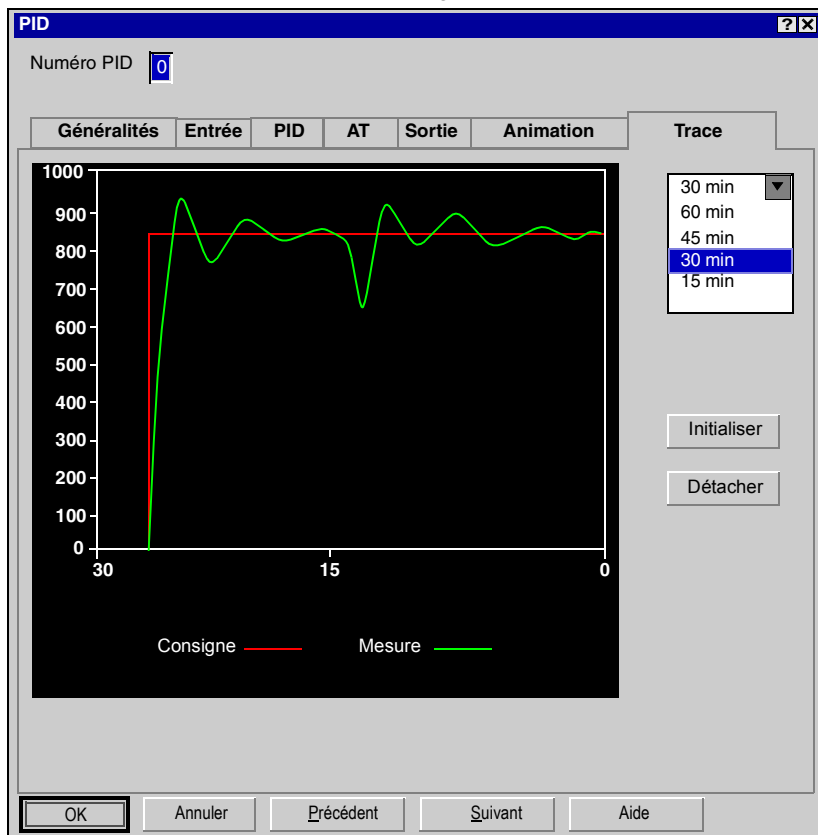
Le tracé des courbes débute dès l'affichage de la fenêtre de mise au point.

<b>Note :</b> Il est accessible en mode connecté.
---

---

## Onglet Animation du PID

L'écran suivant permet de visualiser la régulation du PID.



**Description**

Le tableau suivant décrit les différentes zones de la fenêtre.

Champ	Description
<b>PID numéro</b>	Indiquez ici le numéro du PID que vous désirez visualiser. La valeur est comprise entre 0 et 13, soit 14 PID maximum par application.
<b>Graphe</b>	Cette zone affiche les courbes de la <b>consigne</b> et de la <b>mesure</b> . L'échelle selon l'axe horizontal (X) est déterminée par le menu situé en haut à droite. L'échelle selon l'axe vertical est déterminé par les valeurs de configuration de l'entrée du PID (avec ou sans conversion). Elle est automatiquement optimisée afin de visualiser au mieux les courbes.
<b>Menu d'échelle axe horizontal</b>	Ce menu permet de modifier l'échelle horizontale. Vous pouvez choisir entre 4 valeurs : 15, 30, 45 ou 60 minutes.
<b>Initialiser</b>	Ce bouton efface la courbe et relance la visualisation des tracés.

---

## Etats du PID et codes d'erreurs

### Présentation

Outre la **liste des états du PID** disponible dans la boîte de dialogue **Animation** (voir *Onglet Animation du PID*, p. 479) qui permet de visualiser et commuter vers l'un des 15 derniers états PID, l'automate PID Twido a également la capacité d'enregistrer l'état courant de l'automate PID et du processus AT dans un mot mémoire défini par l'utilisateur.

Pour savoir comment activer et configurer le **mot mémoire de l'état du PID** (%MWi), reportez-vous à l'*Onglet Général du PID*, p. 460.

### Mot mémoire de l'état du PID

Le mot mémoire de l'état du PID peut enregistrer l'un des trois types d'informations du PID, comme suit :

- Etat courant de l'automate PID (état du PID)
- Etat courant du processus d'auto tuning (état AT)
- PID et codes d'erreurs AT

**Note** : Le mot mémoire de l'état PID est en lecture seule.

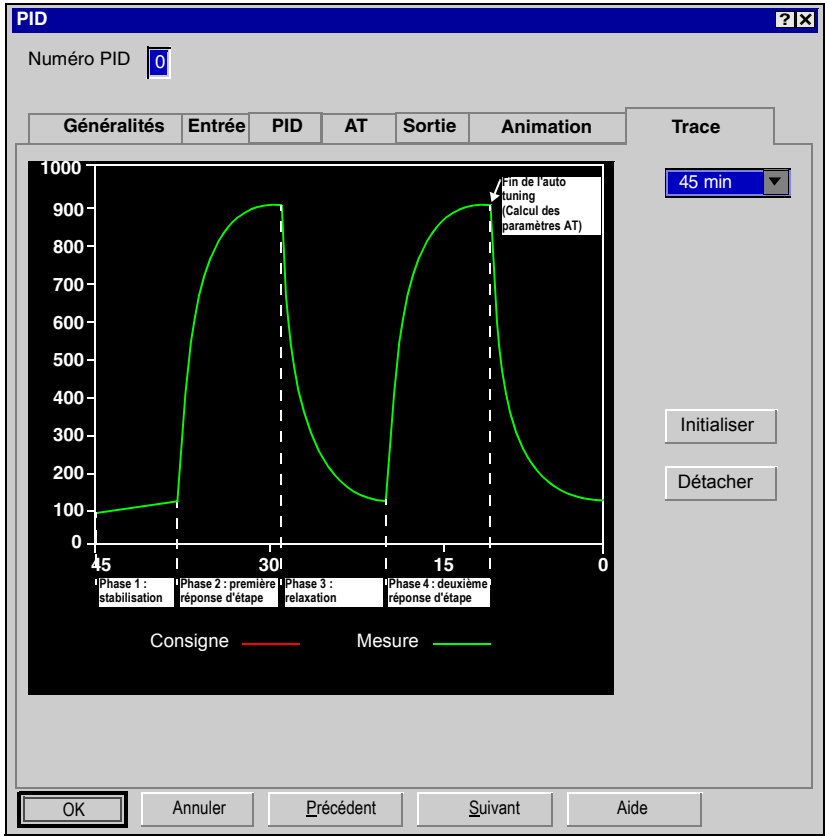
### Mot mémoire de l'état du PID

Le tableau de concordance de codage hexadécimal suivant indique l'état de l'automate PID par rapport au mot mémoire :

Notation hexadécimale de l'état PID	Description
0000h	Le contrôle PID est inactif
2000h	Le contrôle PID est en cours
4000h	La consigne PID est atteinte

Description de l'état AT

Le processus d'auto tuning se divise en 4 phases consécutives. Chaque phase du processus doit être réalisée pour mener à bien l'auto tuning. La courbe de réponse du processus suivante et le tableau décrivent les 4 phases de l'auto tuning PID Twido :



Les phases d'auto tuning sont décrites dans le tableau suivant :

Phase AT	Description
1	<p>La <b>Phase 1</b> est la phase de stabilisation. Elle commence lorsque l'utilisateur lance le processus AT. Au cours de cette phase, l'auto tuning de Twido effectue des vérifications pour garantir que l'état de la variable du processus est stable.</p> <p><b>Remarque :</b> La dernière sortie appliquée au processus avant le début de l'auto tuning est utilisée, tout comme le point de démarrage et le point de relaxation pour le processus d'auto tuning.</p>

Phase AT	Description
2	La <b>Phase 2</b> applique le premier changement d'étape au processus. Elle génère une réponse d'étape au processus similaire à celle illustrée dans le schéma ci-dessus.
3	La <b>Phase 3</b> est la phase de relaxation qui débute lorsque la première réponse d'étape est stabilisée. <b>Remarque</b> : La relaxation se produit vers l'équilibre qui est déterminé comme la dernière sortie appliquée au processus avant le démarrage de l'auto tuning.
4	La <b>Phase 4</b> applique le deuxième changement d'étape au processus avec le même montant et de la même manière que dans la phase 2 décrite ci-dessus. Le processus d'auto tuning se termine et les paramètres AT sont calculés, puis enregistrés dans leurs mots mémoire respectifs une fois la phase 4 réalisée. <b>Remarque</b> : Une fois cette phase exécutée, la variable du processus est restaurée dans le dernier niveau de sortie appliqué au processus avant le démarrage de l'auto tuning.

### Mot mémoire de l'état AT

Le tableau de concordance de codage hexadécimal suivant indique l'état de l'automate PID par rapport au mot mémoire :

Notation hexadécimale de l'état AT	Description
0100h	Phase 1 d'auto tuning en cours
0200h	Phase 2 d'auto tuning en cours
0400h	Phase 3 d'auto tuning en cours
0800h	Phase 4 d'auto tuning en cours
1000h	Processus d'auto tuning terminé

**PID et codes d'erreurs AT**

Le tableau suivant décrit les erreurs d'exécution potentielles pouvant survenir au cours des processus de contrôle PID et d'auto tuning :

Processus PID/AT	Code d'erreur (hexadécimal)	Description
Erreur PID	8001h	Valeur du mode de fonctionnement hors plage
	8002h	Les valeurs minimum et maximum de la conversion linéaire sont égales
	8003h	La limite supérieure pour la sortie numérique est inférieure à la limite inférieure
	8004h	La limite de la variable du processus se trouve en dehors de la plage de conversion linéaire
	8005h	La limite de la variable du processus est inférieure à 0 ou supérieure à 10000
	8006h	La consigne se trouve en dehors de la plage de conversion linéaire
	8007h	La consigne est inférieure à 0 ou supérieure à 10000
	8008h	L'action du contrôle est différente de l'action déterminée au démarrage AT
Erreur d'auto tuning	8009h	Erreur d'auto tuning : la limite de la variable du processus est atteinte
	800Ah	Erreur d'auto tuning : due à un sur-échantillonnage ou à une consigne de sortie trop faible
	800Bh	Erreur d'auto tuning : Kp est égal à zéro
	800Ch	Erreur d'auto tuning : la constante de temps est négative
	800Dh	Erreur d'auto tuning : le retard est négatif
	800Eh	Erreur d'auto tuning : erreur de calcul du paramètre Kp
	800Fh	Erreur d'auto tuning : constante de temps supérieure au délai > 20
	8010h	Erreur d'auto tuning : constante de temps supérieure au délai < 2
	8011h	Erreur d'auto tuning : la limite du paramètre Kp est dépassée
	8012h	Erreur d'auto tuning : la limite du paramètre Ti est dépassée
	8013h	Erreur d'auto tuning : la limite du paramètre Td est dépassée



## Réglage PID avec la fonction d'auto tuning (AT)

### Présentation du réglage PID

La fonction du contrôle PID repose sur les trois paramètres suivants définis par l'utilisateur :  $K_p$ ,  $T_i$  et  $T_d$ . Le réglage PID vise à déterminer de manière précise ces paramètres de processus pour offrir un contrôle optimal du processus.

### Objectif de l'auto tuning

La fonction AT de l'automate Twido est spécifiquement adaptée au réglage automatique des processus thermiques. Etant donné que les valeurs des paramètres PID peuvent varier considérablement d'une régulation à une autre, la fonction d'auto tuning fournie par l'automate Twido peut vous aider à déterminer des valeurs plus précises que celles basées sur vos hypothèses et ce, avec moins d'effort.

### Conditions de l'auto tuning

Lors de l'utilisation de la fonction d'auto tuning, assurez-vous que la régulation et l'automate Twido sont conformes aux quatre exigences suivantes :

- La régulation doit être un système stable à boucle ouverte.
- Au début de l'exécution d'auto tuning, la régulation doit être dans un état stable avec une entrée de processus nulle (par exemple, un four ou un fourneau doit être à température ambiante).
- Lors du fonctionnement de l'auto tuning, veillez à ce qu'aucune perturbation ne s'introduise dans le processus. Sinon, les paramètres calculés seront erronés ou le processus d'auto tuning échouera (par exemple, la porte du four ne doit pas être ouverte, même momentanément).
- Configurez l'automate Twido de façon à ce qu'il effectue une scrutation en **mode périodique**. Une fois que vous avez déterminé la période d'échantillonnage correcte ( $T_s$ ) pour l'auto tuning, la période de scrutation doit être configurée pour que la période d'échantillonnage ( $T_s$ ) soit un multiple exact de la période de scrutation de l'automate Twido.

**Note :** Pour garantir une bonne exécution du contrôle PID et du processus d'auto tuning, il est essentiel de configurer l'automate Twido pour exécuter les scrutations en mode périodique (et non cyclique). En mode périodique, chaque scrutation de l'automate débute à des intervalles réguliers. Ainsi, le taux d'échantillonnage est constant tout au long de la mesure (contrairement au mode cyclique où une scrutation commence dès que la précédente est terminée, ce qui crée un déséquilibre au niveau de la période d'échantillonnage d'une scrutation à l'autre).

## Modes de fonctionnement AT

L'auto tuning peut être utilisé indépendamment (mode AT) ou conjointement avec le contrôle PID (AT + PID) :

- **Mode AT** : Après la convergence du processus AT et une détermination réussie des paramètres  $K_p$ ,  $T_i$  et  $T_d$  du contrôle PID (ou après la détection d'une erreur dans l'algorithme AT), la sortie numérique AT est réglée sur 0 et le message suivant apparaît dans la liste déroulante **Liste des états du PID** : "Auto tuning terminé".
- **Mode AT + PID** : L'auto tuning est lancé en premier. Après l'exécution réussie de l'auto tuning, la boucle du contrôle PID démarre (en fonction des paramètres  $K_p$ ,  $T_i$  et  $T_d$  calculés par l'auto tuning).

**Remarque sur le mode AT+PID** : Si une erreur survient dans l'algorithme AT :

- aucun paramètre PID n'est calculé ;
- la sortie numérique AT est réglée sur la dernière sortie appliquée au processus avant le démarrage de l'auto tuning ;
- un message d'erreur apparaît dans la liste déroulante Liste des états du PID ;
- le contrôle PID est annulé.

### Note : Transition sans à-coups

En **mode AT+PID**, la transition de AT à PID est sans à-coups.

## Méthodes pour déterminer la période d'échantillonnage ( $T_s$ )

Comme il sera expliqué dans les deux sections suivantes (voir *Annexe 1 : Notions fondamentales de la théorie PID*, p. 508 et *Annexe 2 : Premier ordre avec modèle de temporisation* on page 510), la **période d'échantillonnage ( $T_s$ )** est un paramètre clé du contrôle PID. La période d'échantillonnage peut être déduite de la **constante temps AT ( $\tau$ )**.

Il existe deux méthodes pour évaluer la période d'échantillonnage correcte ( $T_s$ ) à l'aide de la fonction d'auto tuning. Ces méthodes sont décrites dans les sections suivantes.

- La méthode de la courbe de réponse du processus
- La méthode des essais et erreurs

Ces deux méthodes sont décrites dans les deux sous-sections suivantes.

**Présentation de la méthode de la courbe de réponse du processus**

Cette méthode consiste à configurer un changement d'étape à l'entrée de régulation et à enregistrer la courbe de sortie du processus par rapport au temps.

La méthode de la courbe de réponse du processus suppose que :

- La régulation peut être décrite de manière adéquate en tant que condition de premier ordre avec modèle de temporisation par la fonction de transfert suivante :

$$\frac{S}{U} = \frac{k}{1 + \tau p} \cdot e^{-\theta p}$$

(Reportez-vous à l'annexe 2 pour obtenir davantage d'informations : Premier ordre avec modèle de temporisation)

---

### Utilisation de la méthode de la courbe de réponse du processus

Pour déterminer la période d'échantillonnage ( $T_s$ ) à l'aide de la méthode de la courbe de réponse du processus, procédez comme suit :

Etape	Action
1	Divers réglages doivent déjà être effectués dans les onglets Général, Entrée, PID, AT et Sortie du PID.
2	Sélectionnez l'onglet <b>PID &gt; Sortie</b> dans le navigateur application.
3	Sélectionnez <b>Autoriser</b> ou <b>Bit adresse</b> dans la liste déroulante <b>Mode manuel</b> pour autoriser la sortie manuelle et définir le champ <b>Sortie</b> sur un niveau élevé (dans la plage [5 000-10 000]).
4	Sélectionnez <b>Automate &gt; Transfert PC =&gt; Automate...</b> dans la barre de menus pour télécharger le programme d'application dans l'automate Twido.
5	Dans la fenêtre de configuration PID, passez en mode <b>Trace</b> .
6	Exécutez le PID et vérifiez l'augmentation de la courbe de réponse.
7	Lorsque la courbe de réponse a atteint un état stable, arrêtez la mesure PID. <b>Remarque</b> : Maintenez la fenêtre PID Trace active.
8	Utilisez la méthode graphique suivante pour déterminer la constante de temps ( $\tau$ ) de la régulation : <ol style="list-style-type: none"> <li>1. Calculez la sortie de la variable du processus pour une augmentation de 63 % (<math>S_{[63\%]}</math>) en utilisant la formule suivante : <math>S_{[63\%]} = S_{[initial]} + (S_{[final]} - S_{[initial]}) \times 63\%</math></li> <li>2. Repérez sur le graphique l'abscisse du temps (<math>t_{[63\%]}</math>) qui correspond à <math>S(63\%)</math>.</li> <li>3. Repérez sur le graphique le temps initial (<math>t_{[initial]}</math>) qui correspond au début de l'augmentation de la réponse du processus.</li> <li>4. Calculez la constante de temps (<math>\tau</math>) de la régulation en utilisant la relation suivante : <math>\tau = t_{[63\%]} - t_{[initial]}</math></li> </ol>
9	Calculez la période d'échantillonnage ( $T_s$ ) basée sur la valeur de ( $\tau$ ) que vous avez déterminé à l'étape précédente, en utilisant la règle suivante : $T_s = \tau / 75$ . <b>Remarque</b> : L'unité de base de la période d'échantillonnage est de 10 ms. Par conséquent, vous devez arrondir la valeur $T_s$ aux 10 ms supérieurs ou inférieurs.
10	Sélectionnez <b>Programme &gt; Editer le mode de scrutation</b> et procédez comme suit : <ol style="list-style-type: none"> <li>1. Définissez le <b>mode de scrutation</b> de l'automate Twido sur <b>Périodique</b>.</li> <li>2. Définissez la <b>période de scrutation</b> de façon à ce que la période d'échantillonnage (<math>T_s</math>) soit un <b>multiple exact</b> de la période de scrutation, en utilisant la règle suivante : Période de scrutation = <math>T_s / n</math>, où "n" est un entier positif.</li> </ol> <b>Remarque</b> : Vous devez choisir "n" pour que la période de scrutation résultante soit un entier positif dans la plage [2 - 150 ms].

**Exemple de  
courbe de  
réponse du  
processus**

Cet exemple vous montre comment mesurer la constante de temps ( $\tau$ ) d'un processus thermique simple à l'aide de la méthode de la courbe du processus décrite dans la sous-section précédente.

Le paramétrage expérimental de la mesure de la constante du temps est le suivant :

- La régulation consiste en un four à air forcé équipé d'un témoin de marche.
- Les mesures de température sont regroupées par l'automate Twido via une sonde Pt100 et la température est enregistrée en °C.
- L'automate Twido contrôle un témoin de marche via la sortie TOR PWM du PID.

L'expérience est réalisée comme suit :

Etape	Action
1	L'onglet <b>Sortie</b> du PID est sélectionné dans la fenêtre de configuration PID.
2	<b>Le mode manuel</b> est sélectionné dans l'onglet Sortie.
3	Le mode manuel <b>Sortie</b> est réglé sur 10 000.
4	Le PID est exécuté depuis l'onglet <b>Trace PID</b> .
5	L'exécution du PID est arrêtée lorsque la température du four est stationnaire.

**Etape** **Action**

6 Les informations suivantes proviennent directement de l'analyse graphique de la courbe de réponse, comme illustré dans le schéma ci-dessous :

The screenshot shows a PID control software window. The main graph displays a step response curve. The y-axis represents the process variable, ranging from 300 to 700. The x-axis represents time, ranging from 60 to 0. The curve starts at a value  $S_{[i]} = 260$  and rises to a steady state value  $S_{[e]} = 660$ . A point on the curve is marked at  $S_{[63\%]} = 512$ , and the time taken to reach this point is indicated as  $\tau = 570 \text{ s}$ . The graph is labeled 'Consigne' (red line) and 'Mesure' (green line). The interface includes tabs for 'Généralités', 'Entrée', 'PID', 'AT', 'Sortie', 'Animation', and 'Trace'. A 'Trace' dropdown menu is set to '60 min'. Buttons for 'Initialiser', 'Retirer', 'OK', 'Annuler', 'Précédent', 'Suivant', and 'Aide' are visible.

Où

- $S_{[i]}$  = valeur initiale de la variable du processus = 260
- $S_{[e]}$  = valeur finale de la variable du processus = 660
- $S_{[63\%]}$  = variable du processus à une augmentation de 63 % =  $S_{[i]} + (S_{[i]} - S_{[e]}) \times 63 \%$   

$$= 260 + (660 - 260) \times 63\%$$

$$= 512$$
- $\tau$  = constante de temps  
 = temps écoulé depuis le début de l'augmentation jusqu'à ce  $S_{[63\%]}$  soit atteint.  

$$= 9 \text{ min } 30 \text{ s} = 570 \text{ s}$$

---

Etape	Action
7	La période d'échantillonnage ( $T_s$ ) est déterminée à l'aide de la relation suivante : $T_s = \tau/75$ $= 570/75 = 7,6 \text{ s (7 600 ms)}$
8	Dans la boîte de dialogue <b>Programme &gt; Editer le mode de scrutation</b> , la <b>période de scrutation</b> doit être définie de sorte que la période d'échantillonnage ( $T_s$ ) est le multiple exact de la période de scrutation, comme dans l'exemple suivant : Période de scrutation = $T_s/76 = 7\,600/76 = 100 \text{ ms}$ (qui satisfait la condition : $2 \text{ ms} \leq \text{période de scrutation} \leq 150 \text{ ms.}$ )

---

**Méthode des essais et erreurs**

La méthode des essais et erreurs consiste à fournir des hypothèses successives de la période d'échantillonnage à la fonction d'auto tuning jusqu'à ce que l'algorithme réussisse à converger vers les paramètres  $K_p$ ,  $T_i$  et  $T_d$  qui sont jugés satisfaisants par l'utilisateur.

**Note :** Contrairement à la méthode de la courbe de réponse du processus, la méthode des essais et erreurs n'est basée sur aucune loi d'approximation de la réponse au processus. Cependant, elle a l'avantage de pouvoir converger vers une valeur de la période d'échantillonnage se trouvant dans le même ordre de grandeur que la valeur actuelle.

Pour effectuer une estimation des essais et erreurs des paramètres d'auto tuning, procédez comme suit :

Etape	Action
1	Sélectionnez l'onglet <b>AT</b> dans la fenêtre de configuration PID.
2	Définissez la <b>limite de sortie</b> d'auto tuning sur <b>10000</b> .
3	Sélectionnez l'onglet <b>PID</b> dans la fenêtre de configuration PID.
4	Saisissez la première ou la $n^{\text{ième}}$ hypothèse dans le champ <b>Période d'échantillonnage</b> . <b>Remarque :</b> Si vous ne possédez pas de première indication sur la plage possible de la période d'échantillonnage, définissez cette valeur sur la plus petite valeur possible : 1 (1 unité de 10 ms).
5	Sélectionnez <b>Automate &gt; Transfert PC =&gt; Automate...</b> dans la barre de menus pour télécharger le programme d'application dans l'automate Twido.
6	Lancez l' <b>auto tuning</b> .
7	Sélectionnez l'onglet <b>Animation</b> dans l'écran de configuration PID.
8	Patiencez jusqu'à la fin du processus d'auto tuning.



Etape	Action
9	<p>Deux cas peuvent se produire :</p> <ul style="list-style-type: none"> <li>● <b>L'auto tuning est réussi</b> : Vous pouvez continuer jusqu'à l'étape 9.</li> <li>● <b>L'auto tuning échoue</b> : Ceci signifie que l'hypothèse courante pour la période d'échantillonnage (<math>T_s</math>) n'est pas correcte. Essayez une nouvelle hypothèse <math>T_s</math> et répétez les étapes 3 à 8, autant de fois que nécessaire jusqu'à ce que le processus d'auto tuning converge.</li> </ul> <p>Suivez ces instructions pour fournir une nouvelle hypothèse <math>T_s</math> :</p> <ul style="list-style-type: none"> <li>● L'auto tuning se termine en indiquant le message d'erreur "<b>La constante de temps calculée est négative</b>" : Ceci signifie que la période d'échantillonnage <b><math>T_s</math> est trop importante</b>. Diminuez la valeur <math>T_s</math> pour fournir une nouvelle hypothèse.</li> <li>● L'auto tuning se termine en indiquant le message d'erreur "<b>Erreur d'échantillonnage</b>" : Ceci signifie que la période d'échantillonnage <b><math>T_s</math> est trop petite</b>. Augmentez la valeur <math>T_s</math> pour fournir une nouvelle hypothèse.</li> </ul>
10	<p>Il se peut que vous ne puissiez pas visualiser les paramètres du contrôle PID (<math>K_p</math>, <math>T_i</math> et <math>T_d</math>) dans l'onglet Animation. Ajustez-les dans l'onglet PID de l'écran de configuration comme il convient.</p> <p><b>Remarque</b> : Si la régulation du PID fournie par cet ensemble de paramètres de contrôle n'indique pas des résultats totalement satisfaisants, vous pouvez affiner l'évaluation des essais et erreurs de la période d'échantillonnage jusqu'à l'obtention d'un ensemble approprié des paramètres de contrôle <math>K_p</math>, <math>T_i</math> et <math>T_d</math>.</p>

### Ajustement des paramètres PID

Pour affiner la régulation du processus fournie par les paramètres PID ( $K_p$ ,  $T_i$ ,  $T_d$ ) provenant de l'auto tuning, vous pouvez également ajuster manuellement la valeur des paramètres, directement à partir de l'onglet PID de l'écran de configuration PID ou via les mots mémoire correspondants (%MW).

**Limites à  
l'utilisation de  
l'auto tuning et  
du contrôle PID**

L'**auto tuning** convient particulièrement aux processus dont la constante de temps ( $\tau$ ) et le temps de retard ( $\theta$ ) respectent l'exigence suivante :  $(\tau + \theta) < 2\,700\text{ s}$  (c'est-à-dire : 45 min)

Le **contrôle PID** convient particulièrement à la régulation des processus qui satisfont à la condition suivante :  $2 < (\tau/\theta) < 20$ , où ( $\tau$ ) est la constante de temps du processus et ( $\theta$ ) le temps de retard.

**Note :** Selon le rapport ( $\tau/\theta$ ) :

- ( $\tau/\theta$ ) < 2 : La régulation PID a atteint ses limites. Des techniques de régulation plus avancées sont requises dans ce cas.
- ( $\tau/\theta$ ) > 20 : Dans ce cas, un simple automate allumé/éteint (ou à deux étapes) peut être utilisé à la place de l'automate PID.

## Résolution des erreurs de la fonction d'auto tuning

Le tableau suivant enregistre les messages d'erreur d'auto tuning et décrit les causes possibles, ainsi que les actions à prendre en matière de résolution :

Message d'erreur	Cause possible	Explication / Solution possible
Erreur d'auto tuning : la limite de la variable du processus est atteinte	La variable du processus a atteint la valeur maximale autorisée.	Est utile pour la sécurité du système. L'auto tuning étant un processus à boucle ouverte, la limite de la variable du processus fonctionne comme une limite supérieure.
Erreur d'auto tuning : due à un sur-échantillonnage ou à une consigne de sortie trop faible	Deux causes possibles : <ul style="list-style-type: none"> <li>la période d'échantillonnage est trop petite,</li> <li>le réglage de la sortie d'auto tuning est trop faible.</li> </ul>	Augmentez la période d'échantillonnage ou la valeur de consigne de sortie d'auto tuning.
Erreur d'auto tuning : la constante de temps est négative	La période d'échantillonnage est peut-être trop importante.	Pour plus d'informations, consultez la section <i>Réglage PID avec la fonction d'auto tuning (AT)</i> , p. 489.
Erreur d'auto tuning : erreur de calcul du paramètre Kp	L'algorithme AT a échoué (pas de convergence).	Vérifiez les paramètres PID et AT et effectuez des ajustements pour améliorer la convergence. Vérifiez également qu'aucune perturbation n'affecte la variable du processus.
Erreur d'auto tuning : constante de temps supérieure au délai > 20	$\tau/\theta > 20$	La régulation PID n'est plus garantie. Pour plus d'informations, consultez la section <i>Réglage PID avec la fonction d'auto tuning (AT)</i> , p. 489.
Erreur d'auto tuning : constante de temps supérieure au délai < 2	$\tau/\theta < 2$	La régulation PID n'est plus garantie. Pour plus d'informations, consultez la section <i>Réglage PID avec la fonction d'auto tuning (AT)</i> , p. 489.
Erreur d'auto tuning : la limite du paramètre Kp est dépassée	La valeur calculée du gain statique (Kp) est supérieure à 10 000.	La sensibilité de la mesure de certaines variables d'application est peut-être trop faible. La plage de mesure de l'application doit être réévaluée dans l'intervalle [0-10 000].
Erreur d'auto tuning : la limite du paramètre Ti est dépassée	La valeur calculée de la constante de temps intégral (Ti) est supérieure à 20 000.	La limite de calcul est atteinte.
Erreur d'auto tuning : la limite du paramètre Td est dépassée	La valeur calculée de la constante de temps dérivative (Td) est supérieure à 10 000.	La limite de calcul est atteinte.

## Méthode de réglage des paramètres PID

---

### Introduction

De nombreuses méthodes de réglages des paramètres d'un PID existent, celle que nous proposons est celle de Ziegler et Nichols qui possède deux variantes :

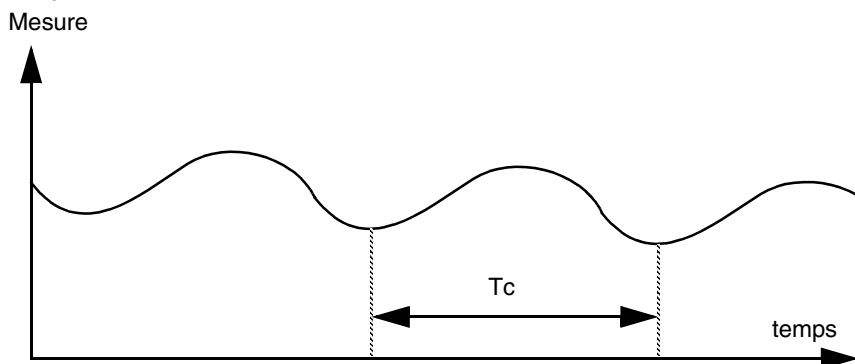
- un réglage en boucle fermée,
- un réglage en boucle ouverte.

Avant de mettre en oeuvre une de ces méthodes, il faut déterminer le sens d'action du PID :

- si une augmentation de la sortie OUT provoque une augmentation de la mesure PV, mettre le PID en inverse ( $K_P > 0$ ),
  - au contraire, si cela provoque une diminution de PV, mettre le PID en direct ( $K_P < 0$ ).
-

## Réglage en boucle fermée

Le principe consiste à utiliser une commande proportionnelle ( $T_i = 0$ ,  $T_d = 0$ ) pour exciter le procédé en augmentant le gain jusqu'à le faire rentrer en oscillation après avoir appliqué un échelon sur la consigne du correcteur PID. Il suffit alors de relever la valeur du gain critique ( $K_{pc}$ ) qui a provoqué l'oscillation non amortie ainsi que la période de l'oscillation ( $T_c$ ) pour en déduire les valeurs donnant un réglage optimal du régulateur.



Selon le type de régulateur (PID ou PI), le réglage des coefficients s'effectue avec les valeurs ci-dessous :

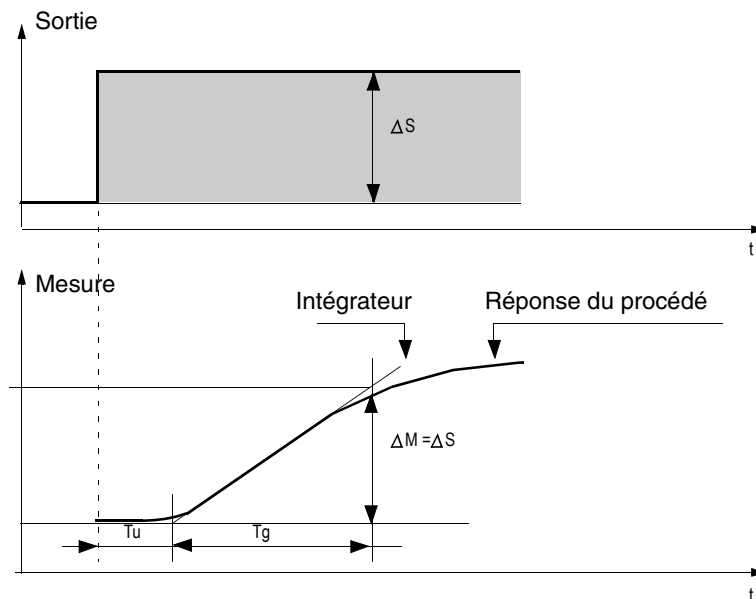
-	$K_p$	$T_i$	$T_d$
PID	$K_{pc}/1,7$	$T_c/2$	$T_c/8$
PI	$K_{pc}/2,22$	$0,83 \times T_c$	-

où  $K_p$  = gain proportionnel,  $T_i$  = temps d'intégration et  $T_d$  = temps de dérivation.

**Note :** Cette méthode de réglage fournit une commande très dynamique pouvant se traduire par des dépassements indésirables lors des changements de points de consigne. Dans ce cas, baisser la valeur du gain jusqu'à obtenir le comportement souhaité.

**Réglage en  
boucle ouverte**

Le régulateur étant en manuel, on applique un échelon sur sa sortie et on assimile le début de la réponse du procédé à un intégrateur avec retard pur..



Le point d'intersection de la droite représentative de l'intégrateur avec l'axe des temps détermine le temps  $T_u$ . On définit ensuite le temps  $T_g$  comme le temps nécessaire à la variable contrôlée (mesure) pour varier de la même amplitude (en % d'échelle) que la sortie du régulateur.

Selon le type de régulateur (PID ou PI), le réglage des coefficients s'effectue avec les valeurs ci-dessous :

-	<b>Kp</b>	<b>Ti</b>	<b>Td</b>
PID	$-1,2 T_g/T_u$	$2 \times T_u$	$0,5 \times T_u$
PI	$-0,9 T_g/T_u$	$3,3 \times T_u$	-

où  $K_p$  = gain proportionnel,  $T_i$  = temps d'intégration et  $T_d$  = temps de dérivation.

**Note :** Attention aux unités. Si le réglage est effectué dans PL7, multiplier par 100 la valeur obtenue pour  $K_P$ .

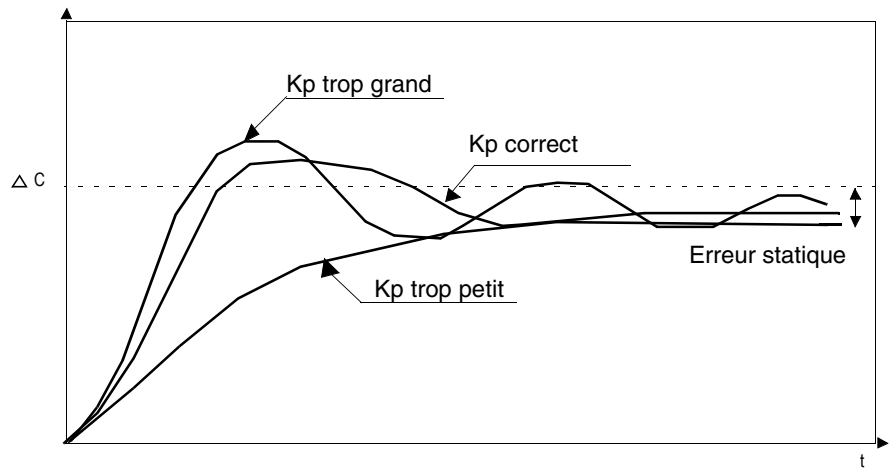
Cette méthode de réglage fournit, elle aussi, une commande très dynamique pouvant se traduire par des dépassements indésirables lors des changements de point de consigne. Dans ce cas, baisser la valeur du gain jusqu'à obtenir le comportement souhaité. L'intérêt de cette méthode réside dans le fait qu'elle ne nécessite aucune hypothèse sur la nature et l'ordre du procédé. Elle s'applique aussi bien aux procédés stables qu'aux procédés réellement intégrateurs. Elle est particulièrement intéressante dans le cas de procédés lents (industrie du verre,...) puisque l'utilisateur n'a besoin que du début de la réponse pour régler les coefficients  $K_p$ ,  $T_i$  et  $T_d$ .

---

## Rôle et influence des paramètres d'un PID

### Influence de l'action proportionnelle

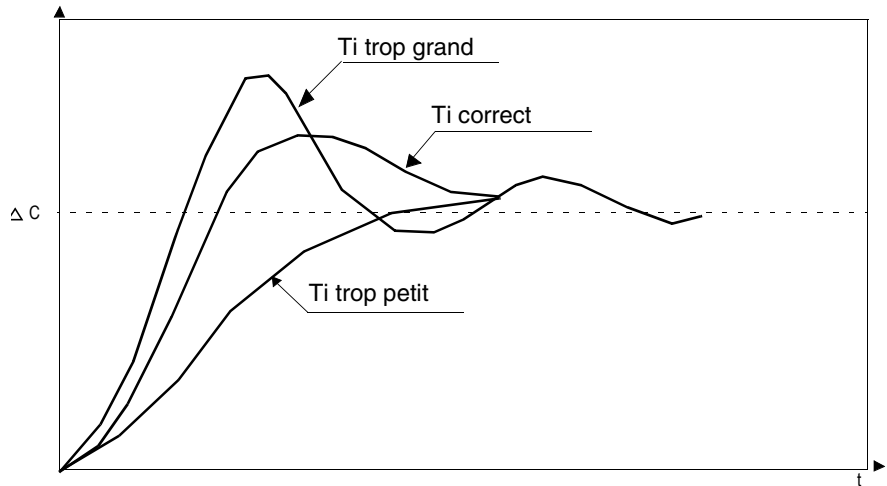
L'action proportionnelle permet de jouer sur la vitesse de réponse du procédé. Plus le gain est élevé, plus la réponse s'accélère, plus l'erreur statique diminue (en proportionnel pur), mais plus la stabilité se dégrade. Il faut trouver un bon compromis entre vitesse et stabilité. L'influence de l'action intégrale sur la réponse du processus à un échelon est la suivante :





### Influence de l'action intégrale

L'action intégrale permet d'annuler l'erreur statique (écart entre la mesure et la consigne). Plus l'action intégrale est élevée ( $T_i$  petit), plus la réponse s'accélère et plus la stabilité se dégrade. Il faut également trouver un bon compromis entre vitesse et stabilité. L'influence de l'action intégrale sur la réponse du processus à un échelon est la suivante :

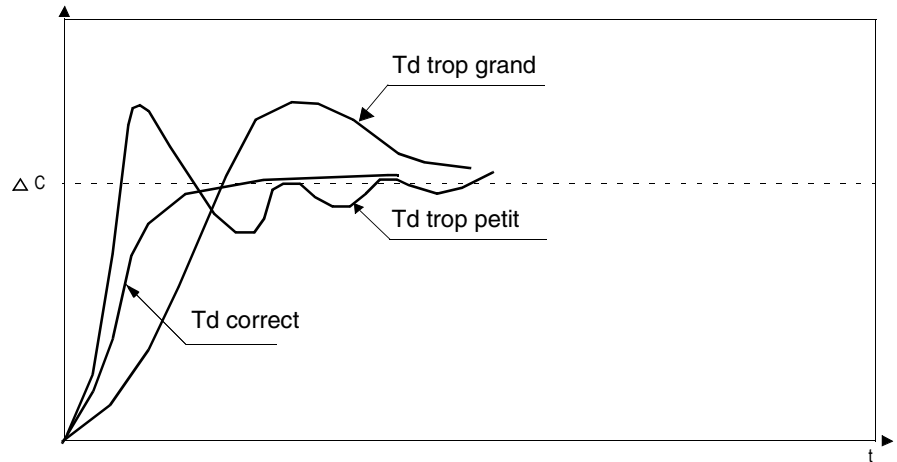


**Note :**  $T_i$  petit signifie une action intégrale élevée.

où  $K_p$  = gain proportionnel,  $T_i$  = temps d'intégration et  $T_d$  = temps de dérivation.

**Influence de  
l'action dérivée**

L'action dérivée est anticipatrice. En effet, elle ajoute un terme qui tient compte de la vitesse de variation de l'écart, ce qui permet d'anticiper en accélérant la réponse du processus lorsque l'écart s'accroît et en le ralentissant lorsque l'écart diminue. Plus l'action dérivée est élevée ( $T_d$  grand), plus la réponse s'accélère. Là encore, il faut trouver un bon compromis entre vitesse et stabilité. L'influence de l'action dérivée sur la réponse du processus à un échelon est la suivante :



## Limites de la régulation PID

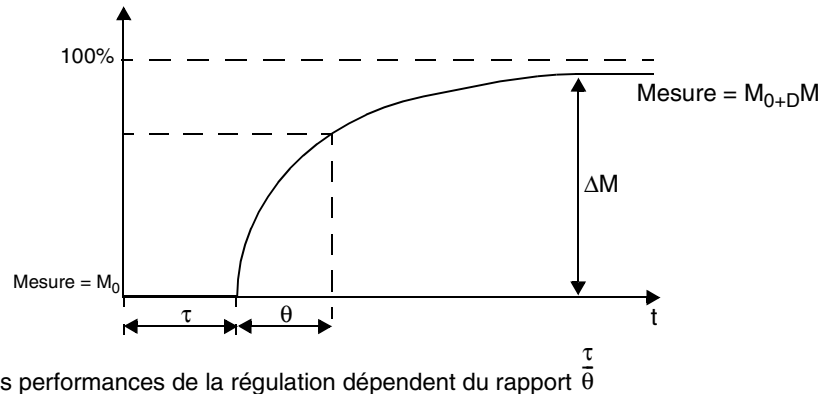
Si on assimile le procédé à un premier ordre à retard pur, de fonction de transfert :

$$(H(p)) = K \frac{(e^{(-\tau)p})}{(1 + \theta p)}$$

avec :

$\tau$  = retard du modèle,

$\theta$  = constante de temps du modèle,



Les performances de la régulation dépendent du rapport  $\frac{\tau}{\theta}$

La régulation PID convient bien dans le domaine suivant :  $2 - \frac{\tau}{\theta} - 20$

Pour  $\frac{\tau}{\theta} < 2$ , c'est à dire des boucles rapides ( $\theta$  petite) ou des procédés à retard important ( $\tau$  grand) la régulation PID ne convient plus, il faut utiliser des algorithmes plus évolués.

Pour  $\frac{\tau}{\theta} > 20$ , une régulation à seuil plus hystérésis suffit.

## Annexe 1 : Notions fondamentales de la théorie PID

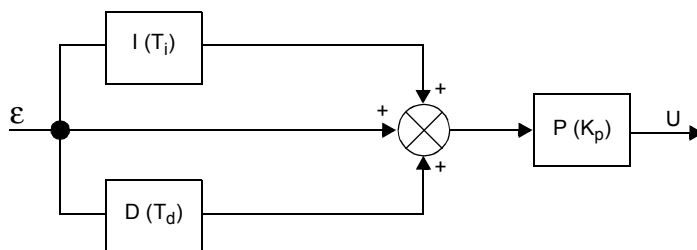
### Introduction

La fonction du contrôle PID intégrée à tous les automates Twido permet de contrôler efficacement les processus industriels simples qui comprennent un déclencheur système (appelé Consigne dans ce document) et une propriété mesurable du système (appelé Mesure ou Variable de régulation).

### Modèle de l'automate PID

L'automate PID Twido implémente une correction PID (série – parallèle) mixte (voir schéma du modèle PID ci-dessous) à partir d'une mesure et d'une consigne analogiques au format [0 – 10 000] et fournit une commande analogique au processus contrôlé analogique au même format.

La forme mixte du modèle de l'automate PID est décrite dans le schéma suivant :



Où

où :

- I = action **intégrale** (agissant indépendamment et parallèle à l'action dérivée),
- D = action **dérivée** (agissant indépendamment et parallèle à l'action intégrale),
- P = action **proportionnelle** (agissant en série sur la sortie associée des actions intégrales et dérivées,
- U = sortie de l'automate PID (alimentation ultérieure comme entrée dans le processus contrôlé.)

## Loi de l'automate PID

L'automate PID comprend une association mixte (série - parallèle) du gain de l'automate (Kp) et des constantes de temps intégrales (Ti) et dérivées (Td). Ainsi, la loi du contrôle PID utilisée par l'automate Twido a la forme suivante (Eq. 1) :

$$u(i) = K_p \cdot \left\{ \varepsilon(i) + \frac{T_s}{T_i} \sum_{j=1}^i \varepsilon(j) + \frac{T_d}{T_s} [\varepsilon(i) - \varepsilon(i-1)] \right\}$$

Où

- Kp = Gain proportionnel de l'automate,
- Ti = Constante de temps intégrale,
- Td = Constante de temps dérivée,
- Ts = Période d'échantillonnage,
- $\varepsilon(i)$  = Ecart ( $\varepsilon(i)$  = consigne – variable de régulation.)

**Note** : Deux différents algorithmes de calcul sont utilisés, selon la valeur de la constante de temps intégrale (Ti) :

- Ti  $\neq$  0 : Dans ce cas, un algorithme incrémentiel est utilisé.
- Ti = 0 : C'est la cas pour les processus de non intégration. Dans ce cas, un algorithme positionnel est utilisé, ainsi qu'un décalage +5 000 appliqué à la variable de sortie PID.

Pour plus de détails sur Kp, Ti et Td, reportez-vous au sous-chapitre *Onglet PID*, p. 466.

Par déduction de (equ. 1) et (equ. 1'), le paramètre clé pour la régulation PID est la **période d'échantillonnage (Ts)**. La période d'échantillonnage dépend étroitement de la **constante de temps ( $\tau$ )**, un paramètre intrinsèque au processus que le PID vise à contrôler. (Voir *Annexe 2 : Premier ordre avec modèle de temporisation*, p. 510.)

## Annexe 2 : Premier ordre avec modèle de temporisation

---

### Introduction

Cette section présente le premier ordre avec le modèle de temporisation utilisé pour décrire différents processus industriels simples mais néanmoins importants, y compris les processus thermiques.

---

### Premier ordre avec modèle de temporisation

On suppose que de simples processus thermiques (à un déclencheur) peuvent se rapprocher de manière adéquate d'un premier ordre avec un modèle de temporisation.

La fonction de transfert de ce processus boucle ouverte de premier ordre a la forme suivante dans le domaine Laplace (*equ.2*) :

$$\frac{S}{U} = \frac{k}{1 + \tau p} \cdot e^{-\theta p}$$

Où

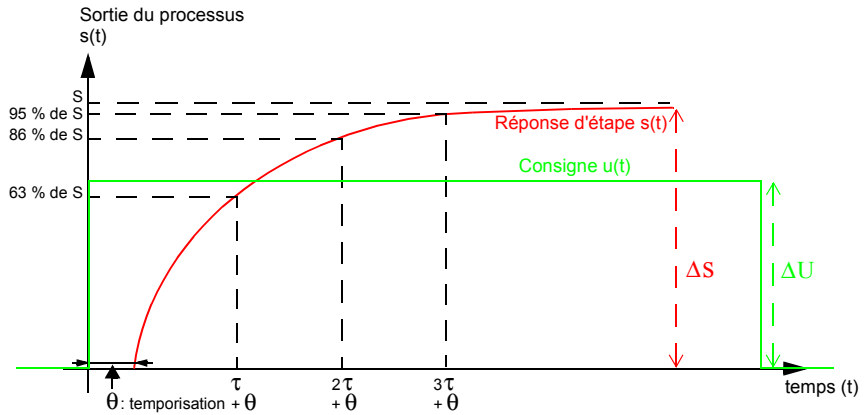
- $k$  = gain statique,
  - $\tau$  = constante de temps,
  - $\theta$  = temps de retard,
  - $U$  = entrée du processus (sortie de l'automate PID),
  - $S$  = sortie du processus.
-

### Constante de temps du processus $\tau$

Le paramètre clé de la loi de réponse du processus (*equ.2*) est la **constante de temps**  $\tau$ . Il s'agit d'un paramètre intrinsèque au processus à contrôler.

La constante de temps ( $\tau$ ) du système de premier ordre est définie par le temps (en secondes) mis par la variable de sortie du système pour atteindre 63 % de la sortie finale à partir du moment où le système commence à réagir au déclenchement de l'étape  $u(t)$ .

Le schéma suivant illustre une réponse à processus de premier ordre type dans le cas d'un déclenchement d'étape :



Où

- $k$  = gain statique calculé comme le ratio  $\Delta S / \Delta U$ ,
- $\tau$  = temps pour une augmentation de 63 % = constante de temps,
- $2\tau$  = temps pour une augmentation de 86 %,
- $3\tau$  = temps pour une augmentation de 95 %.

**Note :** Lorsque l'auto tuning est implémenté, la période d'échantillonnage ( $T_s$ ) doit être choisie dans la plage suivante :  $[\tau/125 < T_s < \tau/25]$ . Il est conseillé d'utiliser  $[T_s = \tau/75]$ . (Voir *Réglage PID avec la fonction d'auto tuning (AT)*, p. 489.)

## 15.4 Instructions sur flottants

---

### Présentation

**Objet de ce sous-chapitre** Ce sous-chapitre décrit les instructions avancées sur flottants (Voir *Objets flottants et mots doubles*, p. 34) du langage TwidoSoft.  
Les instructions de comparaisons et d'affectations sont décrites dans la section *Traitement numérique*, p. 358

---

**Contenu de ce sous-chapitre** Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Instructions arithmétiques sur flottant	513
Instructions Trigonométriques	518
Instructions de conversion	521
Instructions de conversion Entier <-> Flottant	523

---



## Instructions arithmétiques sur flottant

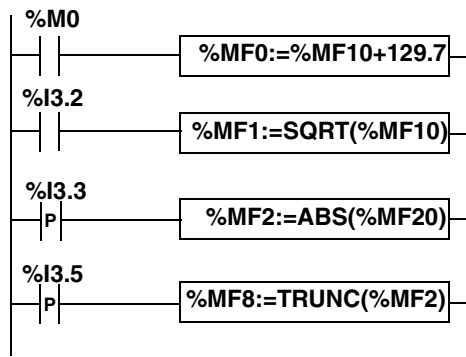
### Généralités

Ces instructions permettent de réaliser une opération arithmétique entre deux opérandes ou sur un opérande.

<b>+</b>	addition de deux opérandes	<b>SQRT</b>	racine carré d'un opérande
<b>-</b>	soustraction de deux opérandes	<b>ABS</b>	valeur absolue d'un opérande
<b>*</b>	multiplication de deux opérandes	<b>TRUNC</b>	partie entière d'une valeur flottante
<b>/</b>	division de deux opérandes	<b>EXP</b>	exponentielle naturelle
<b>LOG</b>	logarithme base 10	<b>EXPT</b>	Puissance d'un entier par un réel
<b>LN</b>	logarithme népérien		

## Structure

## Langage à contacts



## Langage liste d'instructions

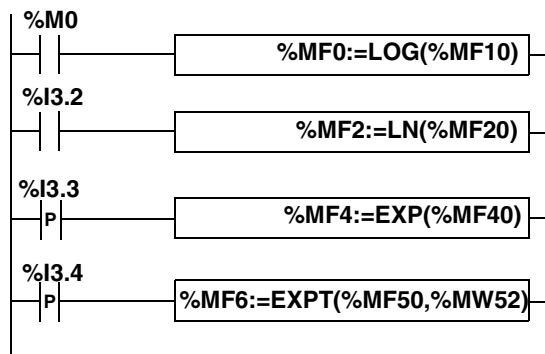
```
LD %M0
[%MF0 := %MF10 + 129.7]
```

```
LD %I3.2
[%MF1 := SQRT (%MF10) ]
```

```
LDR %I3.3
[%MF2 := ABS (%MF20) ]
```

```
LDR %I3.5
[%MF8 := TRUNC (%MF2) ]
```

## Langage à contacts



## Langage liste d'instructions

```
LD %M0
[%MF0:=LOG(%MF10)]

LD %I3.2
[%MF2:=LN(%MF20)]

LDR %I3.3
[%MF4:=EXP(%MF40)]

LDR %I3.4
[%MF6:=EXPT(%MF50,%MW52)]
```

---

**Syntaxe**

## Opérateurs et syntaxe des instructions arithmétiques sur flottant

Opérateurs	Syntaxe
+, -, *, /	Op1:=Op2 Opérateur Op3
SQRT, ABS, TRUNC, LOG, EXP, LN	Op1:=Opérateur(Op2)
EXPT	Op1:=Opérateur (Op2,Op3)

**Note :** Lorsqu'on effectue une addition ou une soustraction entre 2 nombres flottants, les 2 opérandes doivent respecter la condition  $Op1 > Op2 \times 2^{-24}$ , avec  $Op1 > Op2$ . Si cette condition n'est pas respectée le résultat est égal à l'opérande 1 (Op1). Ce comportement est sans grande conséquence lorsqu'ils s'agit d'une opération isolée, puisque l'erreur résultante est très faible ( $2^{-24}$ ), mais a des conséquences inattendues s'il s'agit d'un calcul itératif.

Ex : soit l'instruction **%MF2:= %MF2 + %MF0** répétée indéfiniment. Si les conditions initiales sont %MF0 = 1.0 et %MF2= 0, on observe un blocage de la valeur de %MF2 à 16777216.

Il est donc déconseillé de programmer sans précaution des calculs itératifs. Si on souhaite néanmoins programmer ce type de calcul, il appartient à l'applicatif client de gérer les erreurs de troncature.

## Opérandes des instructions arithmétiques sur flottant:

Opérateurs	Opérande 1 (Op1)	Opérande 2 (Op2)	Opérande 3 (Op3)
+, -, *, /	%MFi	%MFi, %KFi, valeur immédiate	%MFi, %KFi, valeur immédiate
SQRT, ABS, LOG, EXP, LN	%MFi	%MFi, %KFi	[-]
TRUNC	%MFi	%MFi, %KFi	[-]
EXPT	%MFi	%MFi, %KFi	%MWi, %KWi, valeur immédiate

**Règles  
d'utilisation**

- les opérations sur flottants et sur entiers ne peuvent pas être mixées directement. Les opérations de conversion (Voir *Instructions de conversion Entier <-> Flottant*, p. 523) assurent la conversion dans l'un ou l'autre de ces formats.)
- le bit système %S18 est géré de façon identique aux opérations sur entier (Voir *Instructions arithmétiques sur entiers*, p. 367), le mot %SW17 (Voir *Mots système (%SW)*, p. 555) indique la cause du défaut.
- lorsque l'opérande de la fonction est une valeur invalide (exemple : logarithme d'un nombre négatif), elle produit un résultat indéterminé ou infini et fait passer le bit %S18 à 1, le mot %SW17 indique la cause du défaut.

## Instructions Trigonométriques

---

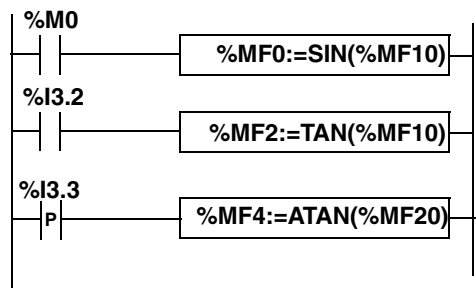
**Généralités** Ces instructions permettent de réaliser des opérations trigonométriques.

<b>SIN</b>	sinus d'un angle exprimé en radian,	<b>ASIN</b>	arc sinus (résultat entre $-\frac{\pi}{2}$ et $\frac{\pi}{2}$ )
<b>COS</b>	cosinus d'un angle exprimé en radian,	<b>ACOS</b>	arc cosinus (résultat entre 0 et $\pi$ )
<b>TAN</b>	tangente d'un angle exprimé en radian.	<b>ATAN</b>	arc tangente (résultat entre $-\frac{\pi}{2}$ et $\frac{\pi}{2}$ )

---

Structure

Langage à contacts



Langage liste d'instructions

```
LD %M0
[%MF0:=SIN(%MF10)]

LD %I3.2
[%MF2:=TAN(%MF10)]

LDR %I3.3
[%MF4:=ATAN(%MF20)]
```

Langage littéral structuré

```
IF %M0 THEN
    %MF0:=SIN(%MF10);
END_IF;
IF %I3.2 THEN
    %MF2:=TAN(%MF10);
END_IF;
IF %I3.3 THEN
    %MF4:=ATAN(%MF20);
END_IF;
```

Syntaxe

Opérateurs, opérandes et syntaxe des instructions opérations trigonométriques:

Opérateurs	Syntaxe	Opérande 1 (Op1)	Opérande 2 (Op2)
SIN, COS, TAN, ASIN, ACOS, ATAN	Op1:=Opérateur(Op2)	%MFi	%MFi, %KFi

**Règles  
d'utilisation**

- lorsque l'opérande de la fonction est une valeur invalide (exemple : arc cosinus d'un nombre supérieur à 1), elle produit un résultat indéterminé ou infini et fait passer le bit %S18 à 1, le mot %SW17 (Voir *Mots système (%SW)*, p. 555) indique la cause du défaut.
  - les fonctions SIN/COS/TAN admettent en paramètre un angle entre  $-4096\pi$  et  $4096\pi$  mais leur précision décroît progressivement pour les angles en dehors de l'intervalle  $-2\pi$  et  $+2\pi$  en raison de l'imprécision apportée par le modulo  $2\pi$  effectué sur le paramètre avant toute opération.
-



## Instructions de conversion

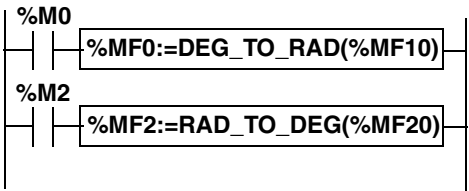
### Généralités

Ces instructions permettent de réaliser des opérations de conversion.

<b>DEG_TO_RAD</b>	conversion de degré en radian, le résultat est la valeur de l'angle compris entre 0 et $2\pi$
<b>RAD_TO_DEG</b>	cosinus d'un angle exprimé en radian, le résultat est la valeur de l'angle compris entre 0 et 360 degrés

### Structure

#### Langage à contacts



#### Langage liste d'instructions

```
LD %M0
[%MF0:=DEG_TO_RAD(%MF10)]

LD %M2
[%MF2:=RAD_TO_DEG(%MF20)]
```

#### Langage littéral structuré

```
IF %M0 THEN
    %MF0:=DEG_TO_RAD(%MF10);
END_IF;
IF %M2 THEN
    %MF2:=RAD_TO_DEG(%MF20);
END_IF;
```

### Syntaxe

Opérateurs, opérandes et syntaxe des instructions de conversion:

Opérateurs	Syntaxe	Opérande 1 (Op1)	Opérande 2 (Op2)
<b>DEG_TO_RAD</b> <b>RAD_TO_DEG</b>	Op1:=Opérateur(Op2)	%MFi	%MFi, %KFi

**Règles  
d'utilisation**

L'angle à convertir doit être compris entre -737280.0 et +737280.0 (pour les conversions DEG\_TO\_RAD) ou entre  $-4096\pi$  et  $4096\pi$  (pour les conversions RAD\_TO\_DEG).  
Pour des valeurs non comprises entre ces bornes le résultat affiché sera + 1.#NAN, les bits %S18 et %SW17:X0 étant positionnés à 1.

---

## Instructions de conversion Entier <-> Flottant

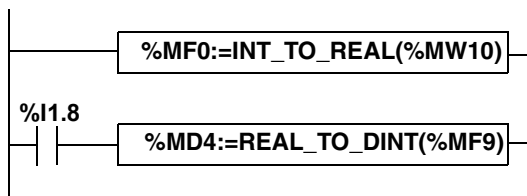
### Généralités

Quatre instructions de conversion sont proposées.  
Liste des instructions de conversion entier<-> flottant :

<b>INT_TO_REAL</b>	conversion d'un mot entier --> flottant
<b>DINT_TO_REAL</b>	conversion double mot entier --> flottant
<b>REAL_TO_INT</b>	conversion flottant --> mot entier (le résultat est la valeur algébrique la plus proche)
<b>REAL_TO_DINT</b>	conversion flottant --> double mot entier (le résultat est la valeur algébrique la plus proche)

### Structure

#### Langage à contacts



#### Langage liste d'instructions

```
LD TRUE
[%MF0:=INT_TO_REAL(%MW10)]
```

```
LD I1.8
[%MD4:=REAL_TO_DINT(%MF9)]
```

#### Langage littéral structuré

```
%MF0:=INT_TO_REAL(%MW10);
IF %I1.8 THEN
  %MD4:=REAL_TO_DINT(%MF9);
END_IF;
```

**Syntaxe**

Opérateurs et syntaxe (conversion d'un mot entier --&gt; flottant) :

Opérateurs	Syntaxe
INT_TO_REAL	Op1=INT_TO_REAL(Op2)

Opérandes (conversion d'un mot entier --&gt; flottant) :

Opérande 1 (Op1)	Opérande 2 (Op2)
%MFi	%MWi,%KWi

**Exemple** : conversion mot entier --> flottant : 147 --> 1.47e+02

Opérateurs et syntaxe (conversion double mot entier --&gt; flottant) :

Opérateurs	Syntaxe
DINT_TO_REAL	Op1=DINT_TO_REAL(Op2)

Opérandes (conversion double mot entier --&gt; flottant) :

Opérande 1 (Op1)	Opérande 2 (Op2)
%MFi	%MDi,%KDi

**Exemple** : conversion mot double entier --> flottant : 68905000 --> 6.8905e+07

Opérateurs et syntaxe (conversion flottant --&gt; mot entier ou mot double entier) :

Opérateurs	Syntaxe
REAL_TO_INT	Op1=Opérateur(Op2)
REAL_TO_DINT	

Opérandes (conversion flottant --&gt; mot entier ou mot double entier) :

Type	Opérande 1 (Op1)	Opérande 2 (Op2)
Mots	%MWi	%MFi, %KFi
Mots doubles	%MDi	%MFi, %KFi

**Exemple** :

conversion flottant --&gt; mot entier : 5978.6 --&gt; 5979

conversion flottant --&gt; mot double entier : -1235978.6 --&gt; -1235979

**Note** : Si lors d'une conversion réel vers entier (ou réel vers entier double mot) la valeur flottante est en dehors des bornes du mot (ou du double mot), le bit %S18 est positionné à 1.

**Précision  
d'arrondi**

La norme IEEE 754 définit 4 modes d'arrondi pour les opérations sur flottant. Le mode utilisé par les instructions ci-dessus est le mode "arrondi au plus près": "si les valeurs représentables les plus proches sont à égale distance du résultat théorique, la valeur fournie sera celle dont le bit de poids faible est égal à 0". Dans certains cas, le résultat de l'arrondi peut donc prendre une valeur par défaut ou une valeur par excès.

Par exemple:

Arrondi de la valeur 10.5 -> 10

Arrondi de la valeur 11.5 -> 12

---

## 15.5 Instructions sur tableaux d'objets

---

### Présentation

#### Objet de ce sous-chapitre

Ce sous chapitre décrit les instructions spécifiques aux tableaux :

- de doubles mots,
- de flottants.

Les instructions d'affectation sur tableaux sont décrites dans le chapitre des "instructions élémentaires" (Voir *Affectation de tables de mots, doubles mots ou flottants*, p. 363).

#### Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Fonction de sommation sur tableaux	527
Fonction de comparaison de tableaux	529
Fonctions de recherche sur tableaux	531
Fonctions de recherche de valeurs maxi et mini sur tableaux	533
Nombre d'occurrences d'une valeur dans un tableau	534
Fonction décalage circulaire sur un tableau	535
Fonction de tri sur tableau	537
Fonction d'interpolation sur tableau de flottants	538
Fonction de moyenne des valeurs d'un tableau de flottants	543

## Fonction de sommation sur tableaux

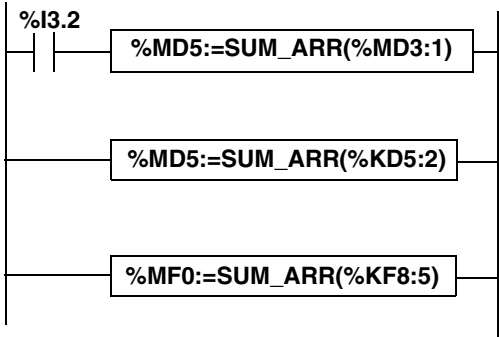
### Généralités

La fonction SUM\_ARR effectue la somme de tous les éléments d'un tableau d'objet :

- si le tableau est constitué de doubles mots, le résultat est donné sous la forme d'un double mot
- si le tableau est constitué de mots flottants, le résultat est donné sous la forme d'un mot flottant

### Structure

#### Langage à contacts



#### Langage liste d'instructions

```
LD %I3.2
[ %MD5 := SUM_ARR ( %MD3 : 1 ) ]
%MD5 := SUM_ARR ( %KD5 : 2 )
%MF0 := SUM_ARR ( %KF8 : 5 )
```

### Syntaxe

Syntaxe de l'instruction de sommation sur tableau:

Res:=SUM\_ARR(Tab)

Paramètres de l'instruction de sommation sur tableau

Type	Résultat (res)	Tableau (Tab)
Tableaux de doubles mots	%MDi	%MDi:L,%KDi:L
Tableaux de flottants	%MFi	%MFi:L,%KFi:L

**Note** : le bit %S18 est mis à 1 lorsque le résultat n'est pas dans les bornes du format double mot suivant l'opérande tableau.

**Exemple**

`%MD5 := SUM ( %MD30 : 4 )`

avec `%MD30=10, %MD31=20, %MD32=30, %MD33=40`

`%MD5=10+20+30+40=100`

---



## Fonction de comparaison de tableaux

### Généralités

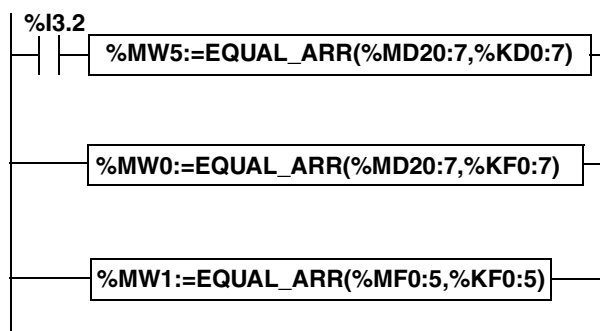
La fonction EQUAL\_ARR effectue la comparaison de 2 tableaux élément par élément.

Si une différence apparaît, le rang des premiers éléments dissemblables est retourné sous forme d'un mot, sinon la valeur retournée est égale à -1.

La comparaison est effectuée sur la totalité du tableau.

### Structure

#### Langage à contacts



#### Langage liste d'instructions

```
LD %I3.2
[%MW5:=EQUAL_ARR(%MD20:7,KD0:7)]
```

#### Langage littéral structuré

```
%MW0:=EQUAL_ARR(%MD20:7,%KF0:7)

%MW1:=EQUAL_ARR(%MF0:5,%KF0:5)
```

Syntaxe

Syntaxe de l'instruction de comparaison de tableaux:

```
Res:=EQUAL_ARR(Tab1,Tab2)
```

Paramètres des instructions de comparaison de tableaux

Type	Résultat (Res)	Tableaux (Tab1 et Tab2)
Tableaux de doubles mots	%MWi	%MDi:L,%KDi:L
Tableaux de flottants	%MWi	%MFi:L,%KFi:L

Note :

- les tableaux doivent être obligatoirement de même longueur et de même type.

Exemple

```
%MW5 :=EQUAL_ARR (%MD30:4 , %KD0:4 )
```

Comparaison des 2 tableaux :

Rang	Tableau de Mots	Tableaux de Constantes	Différence
0	%MD30=10	%KD0=10	=
1	%MD31=20	%KD1=20	=
2	%MD32=30	%KD2=60	Différent
3	%MD33=40	%KD3=40	=

Le mot %MW5 vaut 2 (premier rang différent)

## Fonctions de recherche sur tableaux

### Généralités

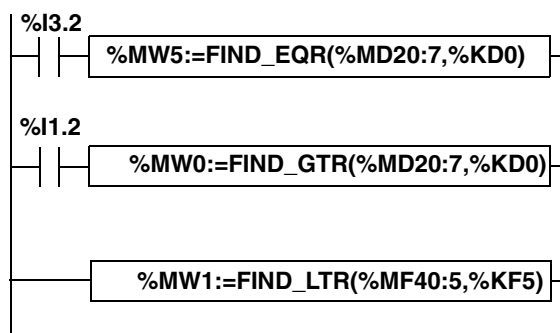
3 fonctions de recherche sont proposées :

- **FIND\_EQR** : recherche de la position dans un tableau de doubles mots ou de flottants du premier élément égal à une valeur donnée
- **FIND\_GTR** : recherche de la position dans un tableau de doubles mots ou de de flottants du premier élément supérieur à une valeur donnée
- **FIND\_LTR** : recherche de la position dans un tableau de doubles mots ou de flottants du premier élément inférieur à une valeur donnée

Le résultat de ces instructions est égal au rang du premier élément trouvé ou à -1 si la recherche est infructueuse.

### Structure

#### Langage à contacts



#### Langage liste d'instructions

```

LD %I3.2
[%MW5:=FIND_EQR(%MD20:7,KD0)]
LD %I1.2
[%MW0:=FIND_GTR(%MD20:7,%KD0)]
%MW1:=FIND_LTR(%MF40:5,%KF5)
  
```

Syntaxe

Syntaxe des instructions de recherche sur tableaux:

Fonction	Syntaxe
FIND_EQR	Res:=Fonction(Tab,Val)
FIND_GTR	
FIND_LTR	

Paramètres des instructions recherche sur tableaux de flottants et doubles mots :

Type	Résultat (Res)	Tableau (Tab)	Valeur (val)
Tableaux de flottants	%MWi	%MFi:L,%KFi:L	%MFi,%KFi
Tableaux de doubles mots	%MWi	%MDi:L,%KDi:L	%MDi,%KDi

Exemple

%MW5 := FIND\_EQR ( %MD30 : 4 , %KD0 )

Recherche de la position du premier double mot =%KD0=30 dans le tableau :

Rang	Tableau de Mots	Résultat
0	%MD30=10	-
1	%MD31=20	-
2	%MD32=30	%MW5=2 (valeur du rang)
3	%MD33=40	-

## Fonctions de recherche de valeurs maxi et mini sur tableaux

### Généralités

2 fonctions de recherche sont proposées :

- **MAX\_ARR** : recherche de la valeur maximum dans un tableau de doubles mots et de flottants
- **MIN\_ARR** : recherche de la valeur minimum dans un tableau de doubles mots et de flottants

Le résultat de ces instructions est égal à la valeur maximum (ou minimum) trouvée dans le tableau.

### Structure

#### Langage à contacts



#### Langage liste d'instructions

```
LD %I1.2
[ %MD0:=MIN_ARR(%MD20:7) ]
%MF8:=MIN_ARR(%MF40:5)
```

### Syntaxe

Syntaxe des instructions de recherche de valeurs maxi et mini sur tableaux:

Fonction	Syntaxe
<b>MAX_ARR</b>	Res:=Fonction(Tab)
<b>MIN_ARR</b>	

Paramètres des instructions de recherche de valeurs maxi et mini sur tableaux:

Type	Résultat (Res)	Tableau (Tab)
Tableaux de doubles mots	%MDi	%MDi:L,%KDi:L
Tableaux de flottants	%MFi	%MFi:L,%KFi:L

# Nombre d'occurrences d'une valeur dans un tableau

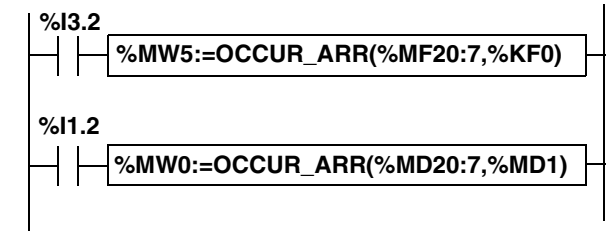
## Généralités

La fonctions de recherche proposée :

- **OCCUR\_ARR** : effectue la recherche dans un tableau de doubles mots ou de flottants du nombre d'éléments égaux à une valeur donnée

## Structure

### Langage à contacts



### Langage liste d'instructions

```
LJ %I3.2
[ %MW5 := OCCUR_ARR ( %MF20 : 7 , %KF0 ) ]
LJ %I1.2
[ %MW0 := OCCUR_ARR ( %MD20 : 7 , %MD1 ) ]
```

## Syntaxe

Syntaxe des instructions de recherche de valeurs maxi et mini sur tableaux:

Fonction	Syntaxe
OCCUR_ARR	Res:=Fonction(Tab,Val)

Paramètres des instructions de recherche de valeurs maxi et mini sur tableaux:

Type	Résultat (Res)	Tableau (Tab)	Valeur (Val)
Tableaux de doubles mots	%MWi	%MDi:L,%KDi:L	%MDi,%KDi
Tableaux de flottants	%MFi	%MFi:L,%KFi:L	%MFi,%KFi

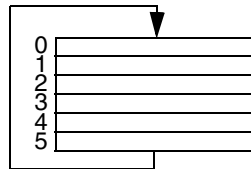
## Fonction décalage circulaire sur un tableau

### Généralités

2 fonctions de décalage sont proposées :

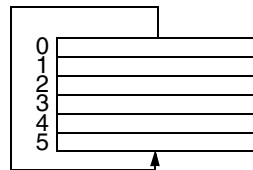
- **ROL\_ARR** : réalise le décalage circulaire de n positions de haut en bas des éléments du tableau de flottants

Illustration des fonctions ROL\_ARR



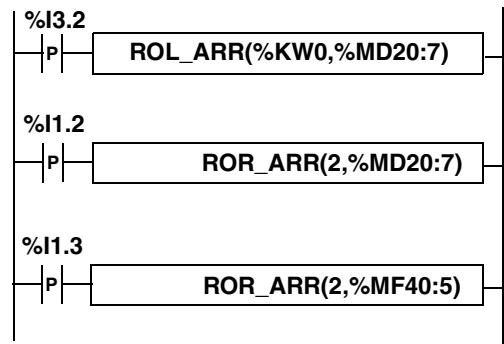
- **ROR\_ARR** : réalise le décalage circulaire de n positions de bas en haut des éléments du tableau de flottants.

Illustration des fonction ROR\_ARR



Structure

Langage à contacts



Langage liste d’instructions

```
LDR %I3.2
[ROL_ARR(%KW0,%MD20:7)]
LDR %I1.2
[ROR_ARR(2,%MD20:7)]
LDR %I1.3
[ROR_ARR(2,%MF40:5)]
```

Syntaxe

Syntaxe des instructions de décalage circulaire sur tableaux de doubles mots ou de flottants **ROL\_ARR** et **ROR\_ARR**

Fonction	Syntaxe
ROL_ARR	Fonction(n,Tab)
ROR_ARR	

Paramètres des instructions de décalage circulaire sur tableaux de flottants: **ROL\_ARR** et **ROR\_ARR** :

Type	Nombre de positions (n)	Tableau (Tab)
Tableaux de flottants	%MWi, valeur immédiate	%MFi:L
Tableaux de doubles mots	%MWi, valeur immédiate	%MDi:L

**Note** : si la valeur de n est négative ou nulle, aucun décalage n'est effectué.



## Fonction de tri sur tableau

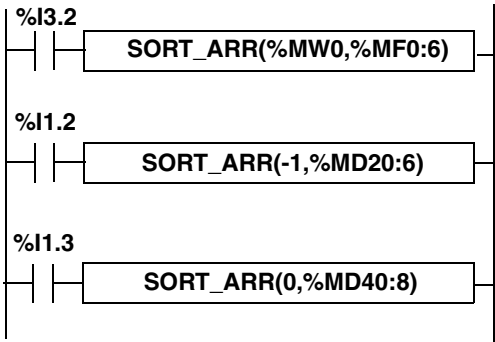
### Généralités

La fonction de tri proposée est la suivante :

- **SORT\_ARR** : réalise les tris par ordre croissant ou décroissant des éléments d'un tableau de doubles mots ou de flottants et range ce qui en résulte dans ce même tableau.

### Structure

#### Langage à contacts



#### Langage liste d'instructions

```
LD %I3.2
[ SORT_ARR (%MW20, %MF0 : 6) ]
LD %I1.2
[ SORT_ARR (-1, %MD20 : 6) ]
LD %I1.3
[ SORT_ARR (0, %MF40 : 8) ]
```

### Syntaxe

Syntaxe des fonctions de tri sur tableaux :

Fonction	Syntaxe
<b>SORT_ARR</b>	Fonction(sens,Tab)

- le paramètre "sens" donne l'ordre du tri: sens > 0 le tri se fait par ordre croissant, sens < 0 le tri s'effectue par ordre décroissant, sens = 0 aucun tri n'est effectué.
- le résultat (tableau trié) est retourné dans le paramètre Tab (tableau à trier).

Paramètres des fonctions de tri sur tableaux :

Type	Sens du tri	Tableau (Tab)
Tableaux de mots doubles	%MWi, valeur immédiate	%MDi:L
Tableaux de flottants	%MWi, valeur immédiate	%MFi:L

## Fonction d'interpolation sur tableau de flottants

### Présentation

La fonction **LKUP** sert à interpoler un ensemble de données flottantes X par rapport à Y pour une valeur X donnée.

### Règle d'interpolation

La fonction LKUP suit la règle d'interpolation linéaire, comme défini dans l'équation suivante :

$$(équation 1 :) \quad Y = Y_i + \left[ \frac{(Y_{i+1} - Y_i)}{(X_{i+1} - X_i)} \cdot (X - X_i) \right]$$

pour  $X_i \leq X \leq X_{i+1}$ , où  $i = 1 \dots (m-1)$  ;

si les valeurs  $X_i$  sont classées par ordre croissant :  $X_1 \leq X_2 \leq \dots X \dots \leq X_{m-1} \leq X_m$ .

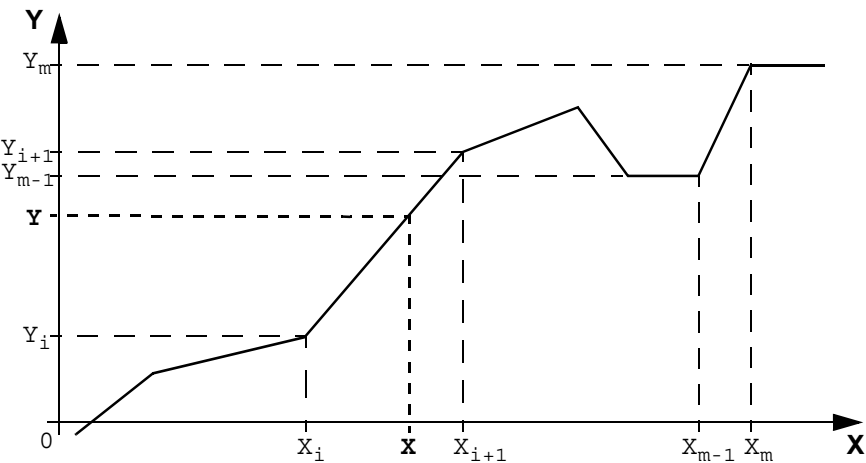
**Note** : Si les deux valeurs  $X_i$  consécutives sont égales ( $X_i = X_{i+1} = X$ ), l'équation (1) fournit une exception invalide. Dans ce cas, pour faire face à cette exception, l'algorithme suivant est utilisé à la place de l'équation (1) :

$$(équation 2 :) \quad Y = \left[ \frac{(Y_{i+1} - Y_i)}{2} \right]$$

pour  $X_i = X_{i+1} = X$ , où  $i = 1 \dots (m-1)$ .

**Représentation graphique de la règle d'interpolation linéaire**

Le graphique suivant illustre la règle d'interpolation linéaire décrite ci-dessus :



**Syntaxe de la fonction LKUP**

La fonction LKUP utilise trois opérandes, dont deux sont des attributs de fonction, comme décrit dans le tableau suivant :

Syntaxe	Opérande 1 (Op1) Variable de sortie	Opérande 2 (Op2) Valeur (X) définie par l'utilisateur	Opérande 3 (Op3) Tableau de variables (X <sub>i</sub> ,Y <sub>i</sub> ) défini par l'utilisateur
[Op1: = LKUP(Op2,Op3)]	%MWi	%MF0	Entier, %MWi ou %KWi

**Définition d'Op1**

**Op1** est le mot mémoire qui contient la variable de sortie de la fonction d'interpolation.

Selon la valeur d'Op1, l'utilisateur peut savoir si l'interpolation a fonctionné ou échoué, ainsi que les causes de l'échec, comme mentionné dans le tableau suivant :

Op1 (%Mwi)	Description
0	Interpolation réussie
1	Erreur d'interpolation : Tableau incorrect, $X_m < X_{m-1}$
2	Erreur d'interpolation : Op2 hors plage, $X < X_1$
4	Erreur d'interpolation : Op2 hors plage, $X > X_m$
8	Taille du tableau des données incorrecte : <ul style="list-style-type: none"><li>● Op3 est défini comme un nombre impair ou</li><li>● <math>Op3 &lt; 6</math>.</li></ul>

<p><b>Note :</b> Op1 <b>ne</b> contient pas la valeur d'interpolation calculée (Y). Pour une valeur (X) donnée, le résultat de l'interpolation (Y) est contenu dans %MF2 du tableau Op3 (voir <i>Définition de Op3</i> ci-dessous).</p>
---

---

**Définition de Op2**

**Op2** est la variable flottante (%MF0 du tableau flottant Op3) qui contient la valeur (X) définie par l'utilisateur et permet de calculer la valeur (Y) interpolée :

- La plage valide pour Op2 est la suivante :  $X_1 \leq Op2 \leq X_m$  .
-

**Définition de Op3** Op3 définit la taille (Op3 / 2) du tableau flottant où les paires de données ( $X_i, Y_i$ ) sont enregistrées.

Les données  $X_i$  et  $Y_i$  sont enregistrées dans des objets flottants avec des index pairs, commençant à %MF4 (notez que les objets flottants %MF0 et %MF2 sont réservés respectivement à la consigne X de l'utilisateur et à la valeur Y interpolée).

Avec un tableau de (m) paires de données ( $X_i, Y_i$ ), l'index supérieur (u) du tableau flottant (%MFu) est défini en utilisant les relations suivantes :

- (équation 3 :)  $Op3 = 2 \cdot m$  ;
- (équation 4 :)  $u = 2 \cdot (Op3 - 1)$  .

La structure du tableau flottant Op3 (%MF<sub>i</sub>) est similaire à celle de l'exemple suivant (où Op3=8) :

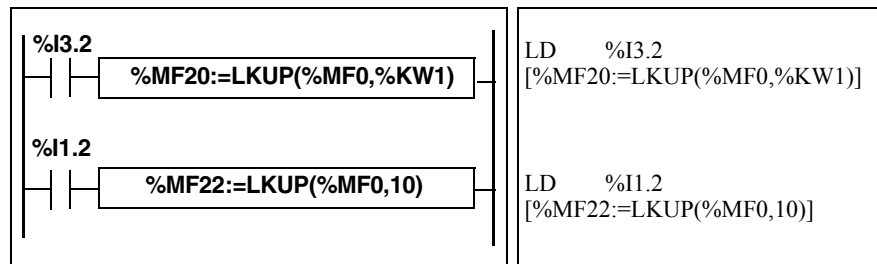
(X)		(X <sub>1</sub> )		(X <sub>2</sub> )		(X <sub>3</sub> )	
%MF0		%MF4		%MF8		%MF12	
	%MF2		%MF6		%MF10		%MF14
	(Y)		(Y <sub>1</sub> )		(Y <sub>2</sub> )		(Y <sub>3</sub> )
							(Op3=8)

**Note** : En raison de la structure du tableau flottant ci-dessus, Op3 doit respecter les exigences suivantes. Sinon, cela déclenche une erreur de la fonction LKUP :

- Op3 est un chiffre pair et
- $Op3 \geq 6$  (2 points de données minimum doivent être disponibles pour permettre une interpolation linéaire).

## Structure

Les opérations d'interpolation sont effectuées de la façon suivante :



**Exemple**

L'exemple suivant illustre l'utilisation d'une fonction d'interpolation LKUP :

```
[%MW20 := LKUP ( %MF0 , 10 ) ]
```

Dans cet exemple :

- %MW20 est Op1 (la variable de sortie).
- %MF0 est la valeur (X) définie par l'utilisateur dont la valeur (Y) correspondante doit être calculée par interpolation linéaire.
- %MF2 enregistre la valeur calculée (Y) générée par l'interpolation linéaire.
- 10 est Op3 (comme indiqué par l'*équation 3* ci-dessus). Il définit la taille du tableau flottant. Élément de classement le plus élevé %MFu, où =18 est indiqué par l'*équation 4*, ci-dessus.

Quatre paires de points de données sont stockées dans le tableau Op3

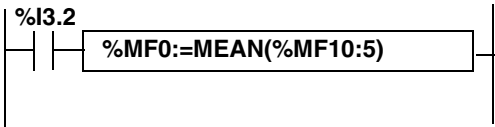
[%MF4..%MF18] :

- %MF4 contient  $X_1$ , %MF6 contient  $Y_1$ .
  - %MF8 contient  $X_2$ , %MF10 contient  $Y_2$ .
  - %MF12 contient  $X_3$ , %MF14 contient  $Y_3$ .
  - %MF16 contient  $X_4$ , %MF18 contient  $Y_4$ .
-

## Fonction de moyenne des valeurs d'un tableau de flottants

**Généralités** La fonction **MEAN** permet de calculer la moyenne des valeurs d'un nombre donné de point d'un tableau de flottants.

**Structure** **Langage à contacts**



**Langage liste d'instructions**

```
LD %I3.2
[ %MF0 := MEAN ( %MF10 : 5 ) ]
```

**Syntaxe** Syntaxe de la fonction de calcul de moyenne d'un tableau de flottants :

Fonction	Syntaxe
MEAN	Result=Fonction(Op1)

Paramètres de la fonction de calcul d'un nombre donné L de valeurs d'un tableau de flottants :

Opérande (Op1)	Résultat (Result)
%MFi:L, %KFi:L	%MFi





---

## Bits système et mots système

# 16

---

### Présentation

#### Objet de ce chapitre

Ce chapitre offre une présentation des bits système et des mots systèmes pouvant être utilisés lors de la création des programmes de régulation d'automates Twido.

#### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Bits système (%S)	546
Mots système (%SW)	555

## Bits système (%S)

---

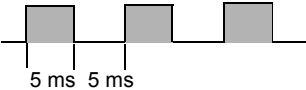
### Introduction

Ce sous-chapitre présente des informations détaillées sur la fonction des bits système, ainsi que sur leur mode de régulation.

---

**Description  
détaillée**

Le tableau suivant présente une description des bits système, ainsi que leur mode de régulation.

Bit système	Fonction	Description	Etat initial	Contrôle
%S0	Démarrage à froid	Normalement à l'état 0, ce bit est réglé sur 1 par : <ul style="list-style-type: none"> <li>• une reprise de l'alimentation avec perte de données (défaillance de la pile) ;</li> <li>• le programme utilisateur ou l'éditeur de tables d'animation ;</li> <li>• l'afficheur.</li> </ul> Ce bit est réglé sur à 1 au cours de la première scrutation. Il est ensuite remis à zéro par le système avant la scrutation suivante.	0	S ou U->S
%S1	Démarrage à chaud	Normalement à l'état 0, ce bit est réglé sur 1 par : <ul style="list-style-type: none"> <li>• une reprise de l'alimentation avec sauvegarde des données ;</li> <li>• le programme utilisateur ou l'éditeur de tables d'animation ;</li> <li>• l'afficheur.</li> </ul> Il est ensuite remis à zéro par le système une fois la scrutation terminée.	0	S ou U->S
%S4 %S5 %S6 %S7	Base temps : 10 ms Base temps : 100 ms Base temps : 1 s Base temps : 1 min	Les changements d'état de ces bits sont cadencés par une horloge interne. Ils ne sont pas synchronisés avec la scrutation de l'automate. Exemple : %S4 	-	S
%S8	Test du câblage	Initialement à l'état 1, ce bit est utilisé pour le test du câblage lorsque l'automate est à l'état "non configuré". Pour modifier la valeur de ce bit, utilisez les touches de l'afficheur pour changer l'état des sorties souhaitées : <ul style="list-style-type: none"> <li>• à l'état 1, mise à zéro des sorties ;</li> <li>• à l'état 0, test du câblage autorisé.</li> </ul>	1	U

Bit système	Fonction	Description	Etat initial	Contrôle
%S9	Mise à zéro des sorties	Normalement à l'état 0. Ce bit peut être réglé sur 1 par le programme ou par le terminal (dans l'éditeur de tables d'animations) : <ul style="list-style-type: none"> <li>à l'état 1, la valeur des sorties est forcée sur 0 lorsque l'automate est en mode d'exécution (RUN) ;</li> <li>à l'état 0, les sorties sont mises à jour normalement.</li> </ul>	0	U
%S10	Défaillance d'E/S	Normalement à l'état 1. Ce bit peut être réglé sur 0 par le système lorsqu'une défaillance d'E/S est détectée.	1	S
%S11	Débordement du chien de garde	Normalement à l'état 0. Ce bit peut être réglé sur 1 par le système lorsque la durée d'exécution du programme (durée de scrutation) dépasse la durée de scrutation maximale (chien de garde logiciel). Le débordement du chien de garde fait passer l'automate en mode suspendu (HALT).	0	S
%S12	Automate en mode d'exécution (RUN)	Ce bit reflète l'état d'exécution de l'automate. Le système met le bit à 1 lorsque l'automate est en cours d'exécution. A l'arrêt, lors de l'initialisation du système et pour tout autre état, ce bit est mis à 0.	0	S
%S13	Premier cycle en mode d'exécution (RUN)	Normalement à l'état 0, ce bit est réglé sur 1 par le système au cours de la première scrutation une fois l'automate passé en mode d'exécution (RUN).	1	S
%S17	Dépassement de capacité	Normalement à l'état 0, ce bit est réglé sur 1 par le système : <ul style="list-style-type: none"> <li>au cours d'une opération de rotation ou de décalage. Le système met la sortie du bit à 1. Doit être testé par le programme utilisateur après chaque opération pouvant provoquer un débordement, puis remis à zéro par l'utilisateur en cas de débordement.</li> </ul>	0	S->U

Bit système	Fonction	Description	Etat initial	Contrôle
%S18	Débordement ou erreur arithmétique	<p>Normalement à l'état 0. Ce bit peut être réglé sur 1 en cas de débordement découlant de l'exécution d'une opération sur 16 bits générant :</p> <ul style="list-style-type: none"> <li>• un résultat supérieur à +32 767 ou inférieur à -32 768, en simple longueur ;</li> <li>• un résultat supérieur à +2 147 483 647 ou inférieur à -2 147 483 648, en double longueur ;</li> <li>• un résultat supérieur à +3,402824E+38 ou inférieur à -3,402824E+38, en flottant ;</li> <li>• une division par 0 ;</li> <li>• la racine carrée d'un nombre négatif ;</li> <li>• une conversion BTI ou ITB non significative : valeur BCD hors plage.</li> </ul> <p>Doit être testé par le programme utilisateur après chaque opération pouvant provoquer un débordement, puis remis à zéro par l'utilisateur en cas de débordement.</p>	0	S->U
%S19	Débordement de la période de scrutation (scrutation périodique)	<p>Normalement à l'état 0, ce bit est réglé sur 1 par le système en cas de débordement d'une période de scrutation (durée de scrutation supérieure à la durée définie par l'utilisateur au moment de la configuration, ou programmée dans %SW0).</p> <p>Ce bit est remis à zéro par l'utilisateur.</p>	0	S->U
%S20	Débordement d'index	<p>Normalement à l'état 0, ce bit est réglé sur 1 lorsque le repère de l'objet indexé devient inférieur à 0 ou supérieur à sa taille maximale.</p> <p>Doit être testé par le programme utilisateur après chaque opération pouvant provoquer un débordement, puis remis à zéro en cas de débordement.</p>	0	S->U
%S21	Initialisation du GRAFCET	<p>Normalement à l'état 0, ce bit est réglé sur 1 par :</p> <ul style="list-style-type: none"> <li>• un redémarrage à froid, %S0=1 ;</li> <li>• le programme utilisateur, uniquement dans la section du programme de prétraitement, à l'aide de l'instruction SET (S %S21) ou d'une bobine SET - (S)- %S21 ;</li> <li>• le terminal.</li> </ul> <p>A l'état 1, il provoque l'initialisation du GRAFCET. Tous les pas actifs sont désactivés et les pas initiaux sont activés.</p> <p>Il est ensuite remis à zéro par le système après l'initialisation du GRAFCET.</p>	0	U->S

Bit système	Fonction	Description	Etat initial	Contrôle
%S22	Remise à zéro du GRAFCET	Normalement à l'état 0, ce bit ne peut être réglé sur 1 par le programme qu'au cours du prétraitement. A l'état 1, il provoque la désactivation des pas de l'ensemble du GRAFCET. Il est remis à zéro par le système au début de l'exécution du traitement séquentiel.	0	U->S
%S23	Prépositionnement et gel du GRAFCET	Normalement à l'état 0, ce bit ne peut être réglé sur 1 par le programme que dans le module du programme de prétraitement. A l'état 1, il valide le prépositionnement du GRAFCET. Le maintien de ce bit sur la valeur 1 a pour effet de geler le GRAFCET (gel du graphe). Il est remis à zéro par le système au début de l'exécution du traitement séquentiel pour garantir l'évolution du GRAFCET à partir de la situation de gel.	0	U->S
%S24	Afficheur	Normalement à l'état 0, ce bit peut être réglé sur 1 par l'utilisateur : <ul style="list-style-type: none"> <li>à l'état 0, l'afficheur fonctionne normalement ;</li> <li>à l'état 1, l'afficheur est gelé, conserve l'affichage courant, le clignotement est désactivé et les touches ne sont plus prises en compte.</li> </ul>	0	U->S
%S31	Masque d'événement	Normalement réglé sur 1 : <ul style="list-style-type: none"> <li>à l'état 0, les événements ne peuvent pas être exécutés et sont mis en attente ;</li> <li>à l'état 1, les événements peuvent être exécutés.</li> </ul> Ce bit peut être réglé sur 0 par l'utilisateur et le système (lors d'un redémarrage à froid).	1	U->S
%S38	Permission des événements à être placés dans la file d'événements	Normalement réglé sur 1 ; <ul style="list-style-type: none"> <li>à l'état 0, les événements ne peuvent pas être placés dans la file d'événements ;</li> <li>à l'état 1, les événements sont placés dans la file d'événements dès qu'ils sont détectés.</li> </ul> Ce bit peut être réglé sur 0 par l'utilisateur et le système (lors d'un redémarrage à froid).	1	U->S
%S39	Saturation de la file d'événements	Normalement réglé sur 0 : <ul style="list-style-type: none"> <li>à l'état 0, tous les événements sont reportés ;</li> <li>à l'état 1, au moins un événement est perdu.</li> </ul> Ce bit peut être réglé sur 0 par l'utilisateur et le système (lors d'un redémarrage à froid).	0	U->S

Bit système	Fonction	Description	Etat initial	Contrôle
%S50	Mise à jour de la date et de l'heure à l'aide des mots %SW49 à %SW53	Normalement à l'état 0, ce bit peut être réglé sur 1 ou 0 par le programme ou l'afficheur : <ul style="list-style-type: none"> <li>à l'état 0, la date et l'heure peuvent être lues ;</li> <li>à l'état 1, la date et l'heure peuvent être mises à jour.</li> </ul> L'horodateur interne de l'automate est mis à jour sur un front descendant de %S50.	0	U->S
%S51	Etat de l'horloge calendaire	Normalement à l'état 0, ce bit peut être réglé sur 1 ou 0 par le programme ou l'afficheur : <ul style="list-style-type: none"> <li>à l'état 0, la date et l'heure sont cohérentes ;</li> <li>à l'état 1, la date et l'heure doivent être initialisées par l'utilisateur.</li> </ul> Lorsque ce bit est réglé sur 1, les données de l'horloge calendaire ne sont pas valides. Il est possible que la date et l'heure n'aient jamais été configurées, que le niveau de la pile soit faible ou que la constante de correction de l'automate ne soit pas valide (jamais configurée, différence entre la valeur d'horloge corrigée et valeur enregistrée ou la valeur hors plage). Le passage de l'état 1 à 0 force l'écriture de la constante de correction sur l'horodateur.	0	U->S
%S52	Horodateur en erreur	Ce bit géré par le système indique que la correction de l'horodateur n'a pas été saisie et que la date et l'heure sont erronées : <ul style="list-style-type: none"> <li>à l'état 0, la date et l'heure sont cohérentes ;</li> <li>à l'état 1, la date et l'heure doivent être initialisées.</li> </ul>	0	S
%S59	Mise à jour de la date et de l'heure à l'aide du mot %SW59	Normalement à l'état 0, ce bit peut être réglé sur 1 ou 0 par le programme ou l'afficheur : <ul style="list-style-type: none"> <li>à l'état 0, le mot système %SW59 n'est pas géré ;</li> <li>à l'état 1, la date et l'heure sont incrémentées ou décrémentées en fonction des fronts montants sur les bits de contrôle réglés dans %SW59.</li> </ul>	0	U
%S66	Affichage du voyant BAT activé/désactivé (uniquement sur les automates qui prennent en charge une pile externe : automates TWDLCA•40DRF)	Ce bit système peut être défini par l'utilisateur. Permet d'allumer/éteindre le voyant BAT : <ul style="list-style-type: none"> <li>à l'état 0, le voyant BAT est allumé (il est remis à zéro par le système lors de la mise sous tension) ;</li> <li>à l'état 1, le voyant BAT est éteint (le voyant reste éteint même si l'alimentation de la pile externe est faible ou si aucune pile n'est présente dans le compartiment à pile).</li> </ul>	0	S ou U->S

Bit système	Fonction	Description	Etat initial	Contrôle
%S69	Affichage du voyant STAT utilisateur	A l'état 0, le voyant STAT est éteint. A l'état 1, le voyant STAT est allumé.	0	U
%S75	Etat de la pile externe (uniquement sur les automates qui prennent en charge une pile externe : automates TWDLCA•40DRF)	Ce bit système est défini par le système. Il indique l'état de la pile externe et peut être lu par l'utilisateur : <ul style="list-style-type: none"> <li>à l'état 0, la pile externe fonctionne normalement ;</li> <li>à l'état 1, l'alimentation de la pile est faible ou la pile ne se trouve pas dans le compartiment à pile.</li> </ul>	0	S
%S95	Restauration des mots mémoire	Ce bit peut être défini lorsque les mots mémoire ont été préalablement enregistrés dans la mémoire EEPROM interne. Le système remet ensuite ce bit à 0 et le nombre de mots mémoire restaurés est défini dans %SW97.	0	U
%S96	Programme de sauvegarde OK	Ce bit peut être lu à n'importe quel moment (soit par le programme ou lors d'un réglage), en particulier après un démarrage à froid ou un redémarrage à chaud : <ul style="list-style-type: none"> <li>à l'état 0, le programme de sauvegarde est incorrect ;</li> <li>à l'état 1, le programme de sauvegarde est correct.</li> </ul>	0	S
%S97	Enregistrement %MW OK	Ce bit peut être lu à n'importe quel moment (soit par le programme ou lors d'un réglage), en particulier après un démarrage à froid ou un redémarrage à chaud : <ul style="list-style-type: none"> <li>à l'état 0, l'enregistrement %MW est incorrect ;</li> <li>à l'état 1, l'enregistrement %MW est correct.</li> </ul>	0	S
%S100	Raccordement du câble de communication TwidoSoft	Indique si le câble de communication TwidoSoft est raccordé : <ul style="list-style-type: none"> <li>à l'état 1, soit le câble de communication TwidoSoft n'est pas raccordé, soit TwidoSoft est connecté ;</li> <li>à l'état 0, le câble de liaison distante TwidoSoft est raccordé.</li> </ul>	-	S
%S101	Changement de repère d'un port (protocole Modbus)	Permet de changer le repère d'un port en utilisant les mots système %SW101 (port 1) et %SW102 (port 2). Pour cela, il faut mettre %S101 à l'état 1. <ul style="list-style-type: none"> <li>A l'état 0, il est impossible de changer le repère. La valeur de %SW101 et %SW102 correspond au repère actuel du port.</li> <li>A l'état 1, il est possible de changer le repère en modifiant les valeurs de %SW101 (port 1) et %SW102 (port 2). Après avoir modifié les valeurs des mots système, il faut remettre %S101 à l'état 0.</li> </ul>	0	U



Bit système	Fonction	Description	Etat initial	Contrôle
%S103 %S104	Utilisation du protocole ASCII	<p>Permet d'utiliser le protocole ASCII sur le port Comm 1 (%S103) ou Comm 2 (%S104). Le protocole ASCII sera configuré à l'aide des mots système %SW103 et %SW105 pour le port Comm 1 et %SW104 et %SW106 pour le port Comm 2.</p> <ul style="list-style-type: none"> <li>● A l'état 0, le protocole utilisé est celui configuré dans TwidoSoft.</li> <li>● A l'état 1, le protocole ASCII est utilisé sur le port Comm 1 (%S103) ou Comm 2 (%S104). Dans ce cas, il faut configurer auparavant les mots système %SW103 et %SW105 pour le port Comm 1 et %SW104 et %SW106 pour le port Comm 2.</li> </ul>	0	U
%S110	Echanges de liaison distante	<p>Ce bit est remis à zéro par le programme ou par le terminal :</p> <ul style="list-style-type: none"> <li>● à l'état 1 pour un maître, tous les échanges de liaison distante (E/S distantes uniquement) sont terminés ;</li> <li>● à l'état 1 pour un esclave, l'échange avec le maître est terminé.</li> </ul>	0	S->U
%S111	Echange de liaison distante unique	<ul style="list-style-type: none"> <li>● A l'état 0 pour un maître, un échange de liaison distante unique est terminé.</li> <li>● A l'état 1 pour un maître, un échange de liaison distante unique est actif.</li> </ul>	0	S
%S112	Connexion de liaison distante	<ul style="list-style-type: none"> <li>● A l'état 0 pour un maître, la liaison distante est activée.</li> <li>● A l'état 1 pour un maître, la liaison distante est désactivée.</li> </ul>	0	U
%S113	Configuration/ fonctionnement de la liaison distante	<ul style="list-style-type: none"> <li>● A l'état 0 pour un maître ou un esclave, la configuration/le fonctionnement de la liaison distante est correct(e).</li> <li>● A l'état 1 pour un maître, la configuration ou le fonctionnement de la liaison distante présente une erreur.</li> <li>● A l'état 1 pour un esclave, la configuration ou le fonctionnement de la liaison distante présente une erreur.</li> </ul>	0	S->U
%S118	Erreur d'E/S distantes	Normalement à l'état 1. Ce bit peut être réglé sur 0 lorsqu'une défaillance d'E/S est détectée sur la liaison distante.	1	S

Bit système	Fonction	Description	Etat initial	Contrôle
%S119	Erreur d'E/S locales	Normalement à l'état 1. Ce bit peut être réglé sur 0 lorsqu'une défaillance d'E/S est détectée sur la liaison distante. %SW118 détermine la nature de la défaillance. Remis à 1 lorsque la défaillance est résolue.	1	S

Description des abréviations utilisées dans le tableau précédent

Tableau des abréviations :

Abréviation	Description
S	Contrôlé par le système
U	Contrôlé par l'utilisateur
U->S	Mis à 1 par l'utilisateur, remis à zéro par le système
S->U	Mis à 1 par le système, remis à zéro par l'utilisateur

## Mots système (%SW)

---

### Introduction

Ce sous-chapitre présente des informations détaillées sur la fonction des mots système, ainsi que sur leur mode de régulation.

---

**Description  
détaillée**

Le tableau suivant fournit des informations détaillées sur la fonction des mots système, ainsi que sur leur mode de régulation.

Mots système	Fonction	Description	Contrôle
%SW0	Période de scrutation de l'automate (tâche périodique)	Modifie la période de scrutation de l'automate, définie lors de la configuration à l'aide du programme utilisateur dans l'éditeur de tables d'animation.	U
%SW6	Etat de l'automate	Etat de l'automate : 0 = NO CONFIG (Non configuré) 2 = STOP (Arrêté) 3 = RUN (Exécution) 4 = HALT (Suspendu)	S

Mots système	Fonction	Description	Contrôle
%SW7	Etat de l'automate	<ul style="list-style-type: none"> <li>● Bit [0] : sauvegarde/restauration en cours : <ul style="list-style-type: none"> <li>● à l'état 1 si une sauvegarde/restauration est en cours ;</li> <li>● à l'état 0 si la sauvegarde/restauration est terminée ou non active.</li> </ul> </li> <li>● Bit [1] : configuration de l'automate correcte : <ul style="list-style-type: none"> <li>● à l'état 1 si la configuration est correcte.</li> </ul> </li> <li>● Bit [3..2] : bits d'état EEPROM : <ul style="list-style-type: none"> <li>● 00 = Pas de cartouche</li> <li>● 01 = Cartouche EEPROM 32 Ko</li> <li>● 10 = Cartouche EEPROM 64 Ko</li> <li>● 11 = Réservé à une utilisation ultérieure</li> </ul> </li> <li>● Bit [4] : application dans RAM différente de l'EEPROM : <ul style="list-style-type: none"> <li>● à l'état 1 si l'application RAM est différente de l'EEPROM.</li> </ul> </li> <li>● Bit [5] : application RAM différente de la cartouche : <ul style="list-style-type: none"> <li>● à l'état 1, si l'application RAM est différente de la cartouche.</li> </ul> </li> <li>● Bit [6] non utilisé (état 0)</li> <li>● Bit [7] : automate réservé : <ul style="list-style-type: none"> <li>● à l'état 1 si réservé.</li> </ul> </li> <li>● Bit [8] : application en mode écriture : <ul style="list-style-type: none"> <li>● à l'état 1 si l'application est protégée.</li> </ul> </li> <li>● Bit [9] non utilisé (état 0)</li> <li>● Bit [10] : port série 2 installé : <ul style="list-style-type: none"> <li>● à l'état 1 si installé.</li> </ul> </li> <li>● Bit [11] : type de port série 2 : (0 = EIA RS-232, 1 = EIA RS-485) : <ul style="list-style-type: none"> <li>● à l'état 0 = EIA RS-232</li> <li>● à l'état 1 = EIA RS-485</li> </ul> </li> <li>● Bit [12] : application valide en mémoire interne : <ul style="list-style-type: none"> <li>● à l'état 1 si l'application est valide.</li> </ul> </li> <li>● Bit [13] : application valide en cartouche : <ul style="list-style-type: none"> <li>● à l'état 1 si l'application est valide.</li> </ul> </li> <li>● Bit [14] : application valide en RAM : <ul style="list-style-type: none"> <li>● à l'état 1 si l'application est valide.</li> </ul> </li> <li>● Bit [15] : prêt pour exécution : <ul style="list-style-type: none"> <li>● à l'état 1 si prêt pour l'exécution.</li> </ul> </li> </ul>	S
%SW11	Valeur du chien de garde logiciel	Contient la valeur maximale du chien de garde (10 à 500 ms) définie par la configuration.	U

Mots système	Fonction	Description	Contrôle
%SW17	Etat par défaut pour opération flottante	Lorsqu'une défaillance est détectée dans une opération arithmétique flottante, le bit %S18 est à l'état 1 et le statut par défaut de %SW17 est mis à jour selon le codage suivant : <ul style="list-style-type: none"> <li>● Bit [0] : opération incorrecte, le résultat n'est pas un nombre (1.#NAN ou -1.#NAN).</li> <li>● Bit 1 : réservé.</li> <li>● Bit 2 : division par 0, le résultat est infini (-1.#INF ou 1.#INF).</li> <li>● Bit 3 : résultat en valeur absolue supérieur à +3,402824e+38, le résultat est infini (-1.#INF ou 1.#INF).</li> </ul>	S et U
%SW18- %SW19	Compteur de temporisation absolu 100 ms	Le compteur fonctionne avec deux mots : <ul style="list-style-type: none"> <li>● %SW18 représente le mot de poids faible.</li> <li>● %SW19 représente le mot de poids fort.</li> </ul>	S et U
%SW30	Durée de la dernière scrutation	Affiche la durée d'exécution du dernier cycle de scrutation de l'automate (en ms). <b>Remarque</b> : Cette durée correspond au temps écoulé entre le début (acquisition des entrées) et la fin (mise à jour des sorties) d'un cycle de scrutation.	S
%SW31	Durée de scrutation maximale	Affiche la durée d'exécution du plus long cycle de scrutation de l'automate (en ms), depuis le dernier démarrage à froid. <b>Remarques</b> : <ul style="list-style-type: none"> <li>● Cette durée correspond au temps écoulé entre le début (acquisition des entrées) et la fin (mise à jour des sorties) d'un cycle de scrutation.</li> <li>● Pour permettre une détection correcte lorsqu'un signal des impulsions est fourni sur l'entrée, la durée d'impulsion (<math>T_{pulsation}</math>) de ce signal doit être supérieure à deux fois la durée de scrutation maximale enregistrée dans le mot système %SW31, comme spécifié par la condition suivante : [<math>T_{pulsation} \geq 2 \times \%SW31</math>].</li> </ul>	S
%SW32	Durée de scrutation minimale	Affiche la durée d'exécution du cycle de scrutation de l'automate le plus court (en ms), depuis le dernier démarrage à froid. <b>Remarque</b> : Cette durée correspond au temps écoulé entre le début (acquisition des entrées) et la fin (mise à jour des sorties) d'un cycle de scrutation.	S
%SW48	Nombre d'événements	Affiche le nombre d'événements exécutés depuis le dernier démarrage à froid. <b>Remarque</b> : A l'état 0 (après chargement de l'application et démarrage à froid), s'incrémente à chaque exécution d'un événement.	S

Mots système	Fonction	Description	Contrôle															
%SW49 %SW50 %SW51 %SW52 %SW53	Horodateur	<div>Fonctions horodateur : mots contenant les valeurs courantes de date et d'heure (en BCD) :</div> <table><tr><td>%SW49</td><td>xN jour de la semaine (N=1 pour lundi)</td></tr><tr><td>%SW50</td><td>00SS secondes</td></tr><tr><td>%SW51</td><td>HHMM heure et minutes</td></tr><tr><td>%SW52</td><td>MMJJ mois et jour</td></tr><tr><td>%SW53</td><td>SSAA siècle et année</td></tr></table> <div>Ces mots sont contrôlés par le système lorsque le bit %S50 est à 0. Ils peuvent être écrits par le programme utilisateur ou par le terminal, lorsque le bit %S50 est réglé sur 1. Sur un front descendant de %S50, l'horodateur interne de l'automate est mis à jour à partir des valeurs écrites dans ces mots.</div>	%SW49	xN jour de la semaine (N=1 pour lundi)	%SW50	00SS secondes	%SW51	HHMM heure et minutes	%SW52	MMJJ mois et jour	%SW53	SSAA siècle et année	S et U					
%SW49	xN jour de la semaine (N=1 pour lundi)																	
%SW50	00SS secondes																	
%SW51	HHMM heure et minutes																	
%SW52	MMJJ mois et jour																	
%SW53	SSAA siècle et année																	
%SW54 %SW55 %SW56 %SW57	Date et heure du dernier arrêt	<div>Mots système contenant la date et l'heure de la dernière coupure secteur ou du dernier arrêt de l'automate (en BCD) :</div> <table><tr><td>%SW54</td><td>SS secondes</td></tr><tr><td>%SW55</td><td>HHMM heure et minutes</td></tr><tr><td>%SW56</td><td>MMJJ mois et jour</td></tr><tr><td>%SW57</td><td>SSAA siècle et année</td></tr></table>	%SW54	SS secondes	%SW55	HHMM heure et minutes	%SW56	MMJJ mois et jour	%SW57	SSAA siècle et année	S							
%SW54	SS secondes																	
%SW55	HHMM heure et minutes																	
%SW56	MMJJ mois et jour																	
%SW57	SSAA siècle et année																	
%SW58	Code du dernier arrêt	<div>Affiche le code indiquant la cause du dernier arrêt :</div> <table><tr><td>1 =</td><td>Front de l'entrée Run/Stop</td><td></td></tr><tr><td>2 =</td><td>Arrêt en cas de défaillance logicielle (dépassement de la scrutation de l'automate)</td><td></td></tr><tr><td>3 =</td><td>Commande d'arrêt</td><td></td></tr><tr><td>4 =</td><td>Coupure secteur</td><td></td></tr><tr><td>5 =</td><td>Arrêt en cas de défaillance matérielle</td><td></td></tr></table>	1 =	Front de l'entrée Run/Stop		2 =	Arrêt en cas de défaillance logicielle (dépassement de la scrutation de l'automate)		3 =	Commande d'arrêt		4 =	Coupure secteur		5 =	Arrêt en cas de défaillance matérielle		S
1 =	Front de l'entrée Run/Stop																	
2 =	Arrêt en cas de défaillance logicielle (dépassement de la scrutation de l'automate)																	
3 =	Commande d'arrêt																	
4 =	Coupure secteur																	
5 =	Arrêt en cas de défaillance matérielle																	

Mot système	Fonction	Description	Contrôle		
%SW59	Réglage de la date courante	Règle la date courante. Contient deux jeux de 8 bits permettant de régler la date courante. L'opération est toujours effectuée sur le front montant du bit. Ce mot est activé par le bit %S59.	U		
		Incrément	Décrément	Paramètre	
		bit 0	bit 8	Jour de la semaine	
		bit 1	bit 9	Secondes	
		bit 2	bit 10	Minutes	
		bit 3	bit 11	Heures	
		bit 4	bit 12	Jours	
		bit 5	bit 13	Mois	
		bit 6	bit 14	Années	
		bit 7	bit 15	Siècles	
%SW60	Correction RTC	Valeur de correction de l'horodateur	U		
%SW63	Code d'erreur du bloc EXCH1	Code d'erreur EXCH1 : 0 - opération réussie 1 - nombre d'octets à émettre trop important (> 250) 2 - table d'émission trop petite 3 - table de mots trop petite 4 - débordement de la table de réception 5 - délai écoulé 6 - émission 7 - mauvaise commande dans la table 8 - port sélectionné non configuré/disponible 9 - erreur de réception 10 - impossible d'utiliser %KW en cas de réception 11 - décalage d'émission plus important que la table d'émission 12 - décalage d'émission plus important que la table de réception 13 - interruption du traitement EXCH par l'automate	S		
%SW64	Code d'erreur du bloc EXCH2	Code d'erreur EXCH2 : Voir %SW63.	S		



Mot système	Fonction	Description	Contrôle
%SW65	Code d'erreur du bloc EXCH3	<p>Le code d'erreur EXCH3 est uniquement implémenté sur les automates Twido TWDLCAE40DRF prenant en charge Ethernet.</p> <p>1-4, 6-13: Voir %SW63. (Remarque : le code d'erreur 5 est incorrect. Il est remplacé par les codes d'erreur 109 et 122 spécifiques à Ethernet qui sont décrits ci-dessous.)</p> <p>Les codes d'erreur suivants sont dédiés à la réponse Modbus :</p> <p>81 - l'automate esclave (serveur) renvoie la réponse FONCTION INCORRECTE</p> <p>82 - l'automate esclave (serveur) renvoie la réponse ADRESSE DE DONNEES INCORRECTE</p> <p>83 - l'automate esclave (serveur) renvoie la réponse VALEUR DE DONNEES INCORRECTE</p> <p>84 - l'automate esclave (serveur) renvoie la réponse ERREUR PERIPHERIQUE ESCLAVE</p> <p>85 - l'automate esclave (serveur) renvoie la réponse CONFIRMATION</p> <p>86 - l'automate esclave (serveur) renvoie la réponse PERIPHERIQUE ESCLAVE OCCUPE</p> <p>87 - l'automate esclave (serveur) renvoie la réponse CONFIRMATION NEGATIVE</p> <p>88 - l'automate esclave (serveur) renvoie la réponse ERREUR DE PARITE MEMOIRE</p> <p>Les codes d'erreur suivants sont spécifiques à Ethernet :</p> <p>101 - aucune adresse IP de ce type</p> <p>102 - la connexion TCP est interrompue</p> <p>103 - aucun socket disponible (toutes les voies de connexion sont occupées)</p> <p>104 - le réseau ne fonctionne pas</p> <p>105 - le réseau est inaccessible</p> <p>106 - le réseau a interrompu la connexion lors de la réinitialisation</p> <p>107 - la connexion a été abandonnée par le poste</p> <p>108 - la connexion a été réinitialisée par le poste</p> <p>109 - délai écoulé pour la connexion</p> <p>110 - rejet de la tentative de connexion</p> <p>111 - l'hôte ne fonctionne pas</p> <p>120 - index inconnu (le périphérique distant n'est pas indexé dans le tableau de configuration)</p> <p>121 - erreur fatale (MAC, puce, adresse IP double) 122 - réception d'avertissement du délai écoulé après l'envoi des données</p> <p>123 - initialisation d'Ethernet en cours</p>	S

<b>Mot système</b>	<b>Fonction</b>	<b>Description</b>	<b>Contrôle</b>
%SW67	Fonction et type d'automate	Contient les informations suivantes : <ul style="list-style-type: none"> <li>● Bits de type d'automate [0 -11]</li> <li>● 8B0 = TWDLC•A10DRF</li> <li>● 8B1 = TWDLC•A16DRF</li> <li>● 8B2 = TWDLMDA20DUK/DTK</li> <li>● 8B3 = TWDLC•A24DRF</li> <li>● 8B4 = TWDLMDA40DUK/DTK</li> <li>● 8B6 = TWDLMDA20DRT</li> <li>● 8B8 = TWDLCAA40DRF</li> <li>● 8B9 = TWDLCAE40DRF</li> <li>● Bit 12, 13, 14 et 15 : non utilisés = 0</li> </ul>	S

<b>Mots système</b>	<b>Fonction</b>	<b>Description</b>	<b>Régulation</b>
%SW73 et %SW74	Etat du système AS-Interface	<ul style="list-style-type: none"> <li>● Bit [0] : à l'état 1 si la configuration est correcte.</li> <li>● Bit [1] : à l'état 1 si l'échange de données est activé.</li> <li>● Bit [2] : à l'état 1 si le module est en mode local.</li> <li>● Bit [3] : à l'état 1 si l'instruction ASI_CMD est terminée.</li> <li>● Bit [4] : à l'état 1 si erreur de l'instruction ASI_CMD en cours.</li> </ul>	S et U
%SW76 à %SW79	Décompteurs 1-4	Ces quatre mots sont utilisés comme temporisateurs à 1 ms. Ils sont décrémentés de manière individuelle par le système, toutes les millisecondes, si leur valeur est positive. Ceci donne quatre décompteurs décomptant en ms (plage de fonctionnement de 1 à 32 767 ms). Le réglage du bit 15 sur 1 permet d'interrompre la décrémentation.	S et U

Mots système	Fonction	Description	Régulation
%SW80	Etat des E/S de base	Bit [0] : voies en fonctionnement normal (pour toutes ses voies) Bit [1] : module en cours d'initialisation (ou initialisation des informations de toutes les voies) Bit [2] : défaut matériel (défaut d'alimentation externe, commun à toutes les voies) Bit [3] : défaut de configuration du module Bit [4] : conversion de la voie d'entrée des données 0 en cours Bit [5] : conversion de la voie d'entrée des données 1 en cours Bit [6] : voie thermocouple d'entrée 0 non configurée Bit [7] : voie thermocouple d'entrée 1 non configurée Bit [8] : non utilisé Bit [9] : non utilisé Bit [10] : voie des données d'entrée analogique 0 au-dessus de la plage Bit [11] : voie des données d'entrée analogique 1 au dessus de la gamme Bit [12] : liaison incorrecte (voie des données d'entrée analogique 0 au-dessous de la plage courante, boucle de courant ouverte) Bit [13] : liaison incorrecte (voie des données d'entrée analogique 1 au-dessous de la plage courante, boucle de courant ouverte) Bit [14] : non utilisé Bit [15] : voie de sortie non disponible	
%SW81	Etat du module d'expansion d'E/S 1 : définitions identiques à %SW80		
%SW82	Etat Module d'expansion d'E/S 2 : définitions identiques à %SW80		
%SW83	Etat Module d'expansion d'E/S 3 : définitions identiques à %SW80		
%SW84	Etat Module d'expansion d'E/S 4 : définitions identiques à %SW80		
%SW85	Etat Module d'expansion d'E/S 5 : définitions identiques à %SW80		
%SW86	Etat Module d'expansion d'E/S 6 : définitions identiques à %SW80		
%SW87	Etat Module d'expansion d'E/S 7 : définitions identiques à %SW80		
%SW81 à %SW87	Etat du module d'expansion		

Mots système	Fonction	Description	Régulation
%SW96	Commande et/ou diagnostic de fonction d'enregistrement et de restauration pour le programme d'application et %MW.	<ul style="list-style-type: none"> <li>● Bit [0] : indique que les mots mémoire %MWi doivent être enregistrés dans l'EEPROM : <ul style="list-style-type: none"> <li>● à l'état 1 si une sauvegarde est requise ;</li> <li>● à l'état 0 si la sauvegarde en cours n'est pas terminée.</li> </ul> </li> <li>● Bit [1] : ce bit est défini par le microprogramme pour indiquer que l'enregistrement est terminé : <ul style="list-style-type: none"> <li>● à l'état 1 si la sauvegarde est terminée ;</li> <li>● à l'état 0 si une nouvelle requête de sauvegarde est demandée.</li> </ul> </li> <li>● Bit [2] : erreur de sauvegarde (reportez-vous aux bits 8, 9, 10 et 14 pour plus d'informations) : <ul style="list-style-type: none"> <li>● à l'état 1 si une erreur est survenue ;</li> <li>● à l'état 0 si une nouvelle requête de sauvegarde est demandée.</li> </ul> </li> <li>● Bit [6] : à l'état 1 si l'automate contient une application vide.</li> <li>● Bit [8] : indique que le nombre de %MW spécifiés dans %SW97 est supérieur au nombre de %MW configurés dans l'application : <ul style="list-style-type: none"> <li>● à l'état 1 si l'erreur est détectée.</li> </ul> </li> <li>● Bit [9] : indique que le nombre de %MW spécifiés dans %SW97 est supérieur au nombre maximum de %MW pouvant être définis par toute application dans TwidoSoft. <ul style="list-style-type: none"> <li>● à l'état 1 si l'erreur est détectée.</li> </ul> </li> <li>● Bit [10] : différence entre la RAM interne et l'EEPROM interne (1 = oui). <ul style="list-style-type: none"> <li>● à l'état 1 s'il y a une différence.</li> </ul> </li> <li>● Bit [14] : indique si une erreur d'écriture sur l'EEPROM s'est produite : <ul style="list-style-type: none"> <li>● à l'état 1 si une erreur est détectée.</li> </ul> </li> </ul>	S et U
%SW97	Commande ou diagnostic de fonction d'enregistrement et de restauration	<p>Lors de la sauvegarde de mots mémoire, cette valeur représente le nombre physique %MW à enregistrer dans l'EEPROM interne. Lors de la restauration de mots mémoire, cette valeur est mise à jour avec le nombre de mots mémoire restaurés dans la RAM.</p> <p>Pour l'opération de sauvegarde, lorsque cette valeur est mise à 0, les mots mémoire ne sont pas stockés. L'utilisateur doit définir le programme de logique utilisateur. Dans le cas contraire, ce programme sera réglé sur 0 dans l'application de l'automate, sauf dans le cas suivant :</p> <p>Lors d'un démarrage à froid, ce mot est réglé sur -1 si l'EEPROM Flash interne ne possède pas de fichier mot mémoire %MW enregistré. Lors d'un démarrage à froid au cours duquel l'EEPROM Flash interne contient une liste de mots mémoire %MW, la valeur du nombre de mots mémoire enregistrés dans le fichier doit être écrite dans le mot système %SW97.</p>	S et U

Mots système	Fonction	Description	Régulation																																
%SW101 %SW102	Valeur de l'adresse Modbus du port	Lorsque le bit %S101 est réglé sur 1, il est possible de modifier l'adresse Modbus du port 1 ou 2. L'adresse du port 1 est %SW101, celle du port 2 est %SW102.	S																																
%SW103 %SW104	Configuration pour l'utilisation du protocole ASCII	<div>Lorsque le bit %S103 (Comm 1) ou %S104 (Comm 2) est réglé sur 1, on utilise le protocole ASCII. Le mot système %SW103 (Comm 1) ou %SW104 (Comm 2) doit être paramétré en fonction des éléments ci-dessous :</div> <table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="8">Fin de la chaîne de caractère</td><td>Bit de données</td><td>Bit d'arrêt</td><td colspan="2">Parité</td><td>RTS / CTS</td><td colspan="3">Débit</td></tr></table> <div><ul style="list-style-type: none"><li>● Débit en bauds :<ul style="list-style-type: none"><li>● 0: 1 200 bauds,</li><li>● 1: 2 400 bauds,</li><li>● 2: 4 800 bauds,</li><li>● 3: 9 600 bauds,</li><li>● 4: 19 200 bauds,</li><li>● 5: 38 400 bauds.</li></ul></li><li>● RTS/CTS :<ul style="list-style-type: none"><li>● 0: désactivé,</li><li>● 1: activé.</li></ul></li><li>● Parité :<ul style="list-style-type: none"><li>● 00: aucune,</li><li>● 10: impaire,</li><li>● 11: paire.</li></ul></li><li>● Bit d'arrêt :<ul style="list-style-type: none"><li>● 0: 1 bit d'arrêt,</li><li>● 1: 2 bits d'arrêt.</li></ul></li><li>● Bit de données :<ul style="list-style-type: none"><li>● 0: 7 bits de données,</li><li>● 1: 8 bits de données.</li></ul></li></ul></div>	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Fin de la chaîne de caractère								Bit de données	Bit d'arrêt	Parité		RTS / CTS	Débit			S
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																				
Fin de la chaîne de caractère								Bit de données	Bit d'arrêt	Parité		RTS / CTS	Débit																						
%SW105 %SW106	Configuration pour l'utilisation du protocole ASCII	<div>Lorsque le bit %S103 (Comm 1) ou %S104 (Comm 2) est réglé sur 1, le protocole ASCII est utilisé. Le mot système %SW105 (Comm 1) ou %SW106 (Comm 2) doit être paramétré en fonction des éléments ci-dessous :</div> <table><tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="8">Trame du délai en ms</td><td colspan="8">Délai de réponse en multiple de 100 ms</td></tr></table>	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Trame du délai en ms								Délai de réponse en multiple de 100 ms								S
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																				
Trame du délai en ms								Délai de réponse en multiple de 100 ms																											

Mots système	Fonction	Description	Régulation
%SW111	Etat de la liaison distante	Indication : le bit 0 correspond à l'automate distant 1, le bit 1 à l'automate distant 2 ... Bit [0] à [6] : <ul style="list-style-type: none"> <li>à l'état 0 = automate distant 1-7 absent</li> <li>à l'état 1 = automate distant 1-7 présent</li> </ul> Bit [8] à bit [14] : <ul style="list-style-type: none"> <li>à l'état 0 = E/S distante détectée sur l'automate distant 1-7</li> <li>à l'état 1 = automate d'extension détecté sur l'automate distant 1-7</li> </ul>	S
%SW112	Code d'erreur de configuration ou de fonctionnement de la liaison distante	00: opérations réussies 01: expiration du délai (esclave) 02: erreur de checksum détectée (esclave) 03: incohérence de configuration (esclave) Défini sur 1 par le système et doit être remis à zéro par l'utilisateur.	S
%SW113	Configuration de la liaison distante	Indication : le bit 0 correspond à l'automate distant 1, le bit 1 à l'automate distant 2... Bit [0] à [6] : <ul style="list-style-type: none"> <li>à l'état 0 = automate distant 1-7 non configuré</li> <li>à l'état 1 = automate distant 1-7 configuré</li> </ul> Bit [8] à bit [14] : <ul style="list-style-type: none"> <li>à l'état 0 = E/S distante configurée en tant qu'automate distant 1-7</li> <li>à l'état 1 = automate d'extension configuré en tant qu'automate distant 1-7</li> </ul>	S
%SW114	Activation des blocs horodateurs	Active ou désactive le fonctionnement des blocs horodateurs, par l'intermédiaire du programme utilisateur ou de l'afficheur. Bit 0 : 1 = active le bloc horodateur n°0 ... Bit 15 : 1 = active le bloc horodateur n°15 Au départ, tous les blocs horodateurs sont activés. Si les blocs horodateurs sont configurés, la valeur par défaut est FFFF. Si aucun bloc horodateur n'est configuré, la valeur par défaut est 0.	S et U
%SW118	Mot d'état de la base automate	Affiche les défaillances détectées sur l'automate maître. Bit 9 : 0 = défaillance ou comm. externe Défaillance Bit 12 : 0 = horodateur non installé Bit 13 : 0 = défaillance de configuration (extension d'E/S configurée, mais absente ou défaillante). Tous les autres bits de ce mot sont à l'état 1 et sont réservés. Pour un automate ne présentant aucune défaillance, la valeur de ce mot est FFFFh.	S

Mots système	Fonction	Description	Régulation
%SW120	Etat de fonctionnement des modules d'expansion d'E/S	Un bit par module. Repère 0 = Bit 0 1 = Mauvaise condition 0 = OK	S

**Description des  
abréviations  
utilisées dans le  
tableau  
précédent**

Tableau des abréviations :

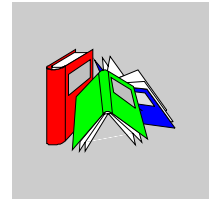
Abréviation	Description
S	Contrôlé par le système
U	Contrôlé par l'utilisateur





---

# Glossaire



!

% Préfixe qui identifie les repères de mémoire interne utilisés dans l'automate pour stocker les valeurs des variables, constantes, E/S, etc. du programme.

---

A

**Adresse IP** Adresse de protocole Internet. Adresse sur 32 bits affectée à des hôtes utilisant TCP/IP.

**Adresse MAC** Adresse de type Media Access Control (contrôle d'accès au support). Il s'agit de l'adresse matérielle d'un périphérique. L'adresse MAC est affectée en usine à un module TCP/IP Ethernet.

**Analyser le programme** Commande permettant de compiler un programme et de rechercher les erreurs qu'il pourrait contenir : erreurs de syntaxe et de structure, symboles sans repère correspondant, ressources non disponibles que le programme tente d'utiliser et taille de programme trop importante pour la capacité de mémoire de l'automate. Les erreurs sont répertoriées dans le visualiseur des erreurs du programme.

**Application** Une application TwidoSoft est composée d'un programme, de données de configuration, de symboles et d'une documentation.

**Arrêter** Commande permettant d'arrêter un programme d'application exécuté par l'automate.

<b>ASCII</b>	(American Standard Code for Information Interchange) Protocole de communication pour représenter les caractères alphanumériques, notamment les lettres, les chiffres et certains caractères graphiques et de contrôle.
<b>Automate</b>	Automate programmable Twido. Il existe deux types d'automate : les automates compacts et les automates modulaires.
<b>Automate compact</b>	Type d'automate Twido fournissant une configuration simple monobloc avec une expansion limitée. Les automates modulaires constituent l'autre type d'automate Twido.
<b>Automate d'extension</b>	Automate Twido configuré en tant qu'esclave sur un réseau de liaison distante. Une application peut être exécutée dans la mémoire de l'automate d'extension et le programme peut accéder aux données d'E/S locales et d'expansion, mais les données d'E/S ne peuvent pas être transmises à l'automate maître. Le programme exécuté dans l'automate d'extension transmet des informations à l'automate maître à l'aide de mots réseau (%INW et %QNW).
<b>Automate distant</b>	Automate Twido configuré pour communiquer avec un automate maître sur un réseau de liaison distante.
<b>Automate maître</b>	Automate Twido configuré en tant que maître sur un réseau de liaison distante.
<b>Automate modulaire</b>	Type d'automate Twido offrant une configuration flexible avec des possibilités d'expansion. Les automates compacts constituent l'autre type d'automate Twido.
<b>Automate programmable</b>	Automate Twido. Il existe deux types d'automate : les automates compacts et les automates modulaires.

---

**B**

<b>Bloc fonction</b>	Unité de programme comportant des entrées et des variables organisées pour calculer les valeurs des sorties à l'aide d'une fonction définie, telle qu'un temporisateur ou un compteur.
<b>Blocs horodateurs</b>	Bloc fonction utilisé pour programmer les fonctions de réglage de la date et de l'heure afin de contrôler les événements. Nécessite l'option Horodateur (RTC).
<b>Bobine</b>	Élément du schéma à contacts représentant une sortie de l'automate.
<b>Bus d'expansion</b>	Permet de connecter les modules d'expansion d'E/S à la base automate.

---

**C**

<b>Cartouche de mémoire</b>	Cartouches de sauvegarde de mémoire en option permettant de sauvegarder et de restaurer une application (données de programme et de configuration). Deux tailles sont disponibles : 32 et 64 Ko.
<b>Chargement automatique</b>	Fonction constamment active permettant de transférer automatiquement une application depuis une cartouche de sauvegarde vers la RAM de l'automate en cas de perte ou d'altération de l'application. A la mise sous tension, l'automate compare l'application se trouvant dans sa RAM avec celle de la cartouche de sauvegarde de mémoire en option (si elle est installée). En cas de différence, l'application de la cartouche de sauvegarde est copiée dans l'automate et dans la mémoire EEPROM interne. Si aucune cartouche de sauvegarde n'est installée, l'application dans la mémoire EEPROM interne est copiée dans l'automate.
<b>Client</b>	Processus informatique nécessitant un service auprès d'autres processus informatiques.
<b>Commentaires</b>	Textes que l'utilisateur saisit afin de donner des informations sur la finalité d'un programme. Pour les programmes en schéma à contacts, vous pouvez saisir jusqu'à trois lignes de texte dans l'en-tête réseau pour décrire la finalité du réseau. Chaque ligne peut contenir entre 1 et 64 caractères. Pour les programmes en liste d'instructions, vous pouvez saisir le texte sur une ligne de programme non numérotée. Les commentaires doivent être insérés entre parenthèses et astérisques, comme suit : (*INSEREZ LES COMMENTAIRES ICI*).
<b>Commutateur</b>	Périphérique réseau connectant au moins deux segments de réseau distincts et permettant ainsi un trafic entre eux. Un commutateur détermine si une trame doit, selon son adresse cible, être bloquée ou transmise.
<b>Compteur</b>	Bloc fonction utilisé pour compter les événements (comptage ou décomptage).
<b>Compteur rapide (VFC)</b>	Bloc fonction proposant une fonction de comptage plus rapide que celle des blocs fonction compteur et compteur rapide (FC). Un compteur rapide (VFC) peut compter à une fréquence maximale de 20 kHz.
<b>Compteurs rapides (FC)</b>	Bloc fonction proposant une fonction de comptage/décomptage plus rapide que celle du bloc fonction compteur. Un compteur rapide (FC) peut compter à une fréquence maximale de 5 kHz.
<b>Concentrateur</b>	Périphérique reliant plusieurs modules souples et centralisés afin de créer un réseau.

<b>Constantes</b>	Valeurs configurées ne pouvant pas être modifiées par le programme en cours d'exécution.
<b>Contact</b>	Élément du schéma à contacts représentant une entrée de l'automate.

---

**D**

<b>Démarrage ou redémarrage à froid</b>	Démarrage de l'automate avec toutes les données initialisées sur les valeurs par défaut, le programme démarrant de zéro avec toutes les variables effacées. Tous les paramètres logiciels et matériels sont initialisés. Le chargement d'une nouvelle application dans la mémoire RAM de l'automate peut provoquer un redémarrage à froid. Un automate sans sauvegarde par pile démarre toujours à froid.
---	---

---

**E**

<b>Editeur de configuration</b>	Fenêtre spécialisée de TwidoSoft permettant de gérer les configurations logicielles et matérielles.
<b>Editeur de langage liste d'instructions</b>	Editeur de programmes simple permettant de créer et d'éditer un programme en liste d'instructions.
<b>Editeur de langage schéma à contacts</b>	Fenêtre TwidoSoft spécialisée permettant d'éditer un programme en schéma à contacts.
<b>Editeur de tables d'animation</b>	Fenêtre spécialisée de l'application TwidoSoft permettant de visualiser et de créer des tables d'animation.
<b>EEPROM</b>	Mémoire morte effaçable et programmable électriquement (de l'anglais Electrically Erasable Programmable Read-Only Memory). Twido est doté d'une mémoire EEPROM interne et d'une cartouche de mémoire EEPROM externe en option.
<b>Effacer</b>	Commande permettant de supprimer l'application dans l'automate, elle comporte deux options : <ul style="list-style-type: none"><li>● Efface le contenu de la RAM de l'automate, de l'EEPROM interne de l'automate et de la cartouche de sauvegarde en option installée.</li><li>● Efface uniquement le contenu de la cartouche de sauvegarde en option installée.</li></ul>

---

<b>En-tête réseau</b>	Panneau apparaissant directement sur un réseau de schéma à contacts et pouvant être utilisé pour donner des informations sur la finalité de celui-ci.
<b>Entrée à mémorisation d'état</b>	Les impulsions entrantes sont capturées et enregistrées afin d'être analysées ultérieurement par l'application.
<b>Etape</b>	Une étape Grafcet désigne un état du fonctionnement séquentiel de l'automate.
<b>Etat connecté</b>	Etat de fonctionnement de TwidoSoft qui est affiché dans la barre d'état lorsqu'un PC est connecté à un automate.
<b>Etat initial</b>	Etat de fonctionnement de TwidoSoft affiché dans la barre d'état lorsque TwidoSoft est démarré ou qu'aucune application n'est ouverte.
<b>Etat local</b>	Etat de fonctionnement de TwidoSoft qui est affiché dans la barre d'état lorsque aucun PC n'est connecté à un automate.
<b>Etat Surveillance</b>	Etat de fonctionnement de TwidoSoft qui est affiché dans la barre d'état lorsqu'un PC est connecté à un automate dans un mode sans écriture.
<b>Etats de fonctionnement</b>	Indique l'état de TwidoSoft. Affiché dans la barre d'état. Il existe quatre états de fonctionnement : initial, local, connecté et surveillance.
<b>Exécuter</b>	Commande permettant d'exécuter un programme d'application sur l'automate.
<b>Executive Loader</b>	Application Windows 32 bits permettant de télécharger un nouveau microprogramme de l'automate vers un automate Twido.

---

**F**

<b>Fichier d'application</b>	Les applications Twido sont enregistrées dans des fichiers portant l'extension .twd.
<b>FIFO</b>	Premier entré, Premier sorti (de l'anglais First In, First Out). Bloc fonction permettant de mettre les opérations en file d'attente.
<b>Fonctions Date/Heure</b>	Fonctions permettant de contrôler les événements par mois, jour et heure. Voir "Blocs horodateurs".

**Forçage** Attribution volontaire des valeurs 0 et 1 aux entrées et sorties de l'automate, même si les valeurs réelles sont différentes. Permet de déboguer un programme pendant son animation.

---

## G

**Gestionnaire de ressources** Composant de TwidoSoft qui surveille les besoins en mémoire d'une application lors de la programmation et de la configuration, en suivant les références aux objets logiciels faites par une application. Un objet est considéré comme étant référencé par l'application lorsqu'il est utilisé comme opérande dans une instruction de langage liste d'instructions ou dans un réseau de schéma à contacts. Affiche les informations d'état relatives au pourcentage de mémoire totale utilisée et émet un avertissement si l'espace mémoire disponible est insuffisant. Voir "Indicateur d'utilisation de la mémoire".

**Grafcet** Permet de représenter graphiquement et de façon structurée le fonctionnement d'un automatisme séquentiel.  
Il s'agit d'une méthode analytique qui divise toute régulation d'automatisation en une série d'étapes auxquelles des actions, des transitions et des conditions sont associées.

---

## H

**Horodateur** Option permettant de maintenir une horloge à l'heure pendant une durée déterminée lorsque l'automate n'est pas sous tension.

**Hôte** Nœud d'un réseau.

---

## I

**Indicateur d'utilisation de la mémoire** Section de la barre d'état de la fenêtre principale de TwidoSoft qui affiche le pourcentage d'utilisation par une application de la mémoire totale de l'automate. Emet un avertissement lorsque l'espace mémoire disponible est insuffisant.

**Initialiser** Commande qui rétablit les états initiaux de toutes les valeurs des données. L'automate doit être en mode Arrêt ou Erreur.

---

---

<b>Instance</b>	Dans un programme, objet unique qui appartient à un type précis de bloc fonction. Par exemple, dans le format de temporisateur %T <sub>Mi</sub> , i est un nombre qui représente l'instance.
<b>Instructions réversibles</b>	Méthode de programmation permettant de visualiser les instructions alternativement comme des instructions de liste d'instructions ou des réseaux de schéma à contacts.
<b>Internet</b>	Interconnexion mondiale de réseaux de communication par ordinateur fonctionnant sur TCP/IP.
<b>IP</b>	Protocole Internet (Internet Protocol). Protocole classique de la couche réseau. IP est le plus souvent utilisé avec TCP.

---

**L**

<b>Langage liste d'instructions</b>	Un programme écrit en langage liste d'instructions (IL) consiste en une série d'instructions exécutées de manière séquentielle par l'automate. Chaque instruction comprend un numéro de ligne, un code d'instruction et un opérande.
<b>Langage schéma à contacts</b>	Un programme écrit en langage schéma à contacts consiste en la représentation graphique d'instructions d'un programme de l'automate, avec des symboles pour les contacts, bobines et blocs, sous la forme d'une série de réseaux exécutés de manière séquentielle par un automate.
<b>Liaison distante</b>	Bus maître/esclave à haut débit conçu pour assurer l'échange d'une petite quantité de données entre un automate maître et un maximum de sept automates distants (esclaves). Deux types d'automate distant peuvent être configurés pour transférer des données vers un automate maître : un automate d'extension pour transférer les données d'application et un automate d'E/S distant pour transférer les données d'E/S. Un réseau de liaison distante peut comprendre des automates des deux types.
<b>LIFO</b>	Dernier entré, Premier sorti (de l'anglais Last In, First Out). Bloc fonction permettant d'effectuer des opérations de pile.
<b>Lignes de commentaire</b>	Dans les programmes en liste d'instructions, les commentaires peuvent être saisis sur des lignes distinctes des instructions. Les lignes de commentaires ne portent pas de numéro de ligne et doivent être insérées entre parenthèses et astérisques, comme suit : (*INSEREZ LES COMMENTAIRES ICI*).

---

**M**

<b>Masque de sous-réseau</b>	Masque de bit permettant d'identifier ou de déterminer les bits de l'adresse IP correspondant à l'adresse réseau et les bits correspondant aux portions du sous-réseau de l'adresse. Le masque de sous-réseau est constitué de l'adresse réseau et des bits réservés à l'identification du sous-réseau.
<b>MBAP</b>	Protocole d'application Modbus (de l'anglais Modbus Application Protocol).
<b>Microprogramme de l'automate</b>	Système d'exploitation exécutant les applications et gérant les opérations de l'automate.
<b>Modbus</b>	Protocole de communication maître-esclave permettant à un maître unique d'obtenir des réponses des esclaves.
<b>Mode connecté</b>	Mode de fonctionnement de TwidoSoft dans lequel un PC est connecté à l'automate et dans lequel l'application contenue dans la mémoire du PC est identique à celle contenue dans la mémoire de l'automate. Le fonctionnement en ligne permet de déboguer une application.
<b>Mode de scrutation</b>	Indique la façon dont l'automate scrute un programme. Il existe deux types de scrutation : le mode normal (cyclique), dans lequel la scrutation s'effectue en permanence, ou le mode périodique, dans lequel la scrutation ne s'effectue que pendant une durée limitée (dans une plage de 2 à 150 ms) avant de lancer la scrutation suivante.
<b>Mode local</b>	Mode de fonctionnement de TwidoSoft dans lequel aucun PC n'est connecté à l'automate et dans lequel l'application contenue dans la mémoire du PC est différente de celle contenue dans la mémoire de l'automate. Le mode local permet de créer et de développer une application.
<b>Modules d'expansion d'E/S</b>	Les modules d'expansion d'E/S en option sont disponibles pour ajouter des points d'E/S à un automate Twido. (Certains modèles d'automate ne prennent pas en charge l'expansion).

---



---

**N**

<b>Navigateur d'application</b>	Fenêtre spécialisée de l'application TwidoSoft qui affiche l'arborescence graphique d'une application. Facilite l'affichage et la configuration d'une application.
<b>Nœud</b>	Périphérique adressable sur un réseau de communication.

---

**O**

<b>Opérande</b>	Nombre, repère ou symbole représentant une valeur qu'un programme peut manipuler dans une instruction.
<b>Opérateur</b>	Symbole ou code indiquant l'opération qu'une instruction doit réaliser.

---

**P**

<b>Paquet</b>	Unité de données envoyée sur un réseau.
<b>Passerelle</b>	Périphérique reliant des réseaux dont l'architecture est différente et fonctionnant sur la couche application. Ce terme peut faire référence à un routeur.
<b>Passerelle par défaut</b>	Adresse IP du réseau ou de l'hôte vers laquelle sont envoyés tous les paquets adressés à un réseau ou à un hôte inconnu. La passerelle par défaut est généralement un routeur ou un autre périphérique.
<b>PC</b>	Ordinateur personnel (de l'anglais Personal Computer).
<b>PLS</b>	Générateur d'impulsions. Bloc fonction qui génère une onde carrée avec des cycles d'activité de 50 % et d'inactivité de 50 %.
<b>Point de réglage analogique</b>	Tension appliquée qui peut être réglée et convertie en une valeur numérique utilisable par une application.
<b>Préférences</b>	Boîte de dialogue comprenant des options sélectionnables permettant de configurer les éditeurs de programmes en liste d'instructions et schéma à contacts.

<b>Programmateur cyclique</b>	Bloc fonction dont le fonctionnement est semblable à celui des programmeurs cycliques électromécaniques : les modifications d'étapes sont associées aux événements externes.
<b>Protection</b>	Se réfère aux deux types de protection d'une application : la protection par mot de passe, qui permet de contrôler l'accès à l'application, et la protection de l'application de l'automate, qui empêche la lecture et l'écriture sur un programme d'application.
<b>Protocole</b>	Définit les formats de message et un jeu de règles utilisé par au moins deux périphériques pour communiquer à l'aide de ces formats.
<b>PWM</b>	Modulation de largeur (de l'anglais Pulse Width Modulation). Bloc fonction qui génère une onde rectangulaire avec un cycle d'activité variable pouvant être défini par un programme.

---

**R**

<b>RAM</b>	De l'anglais Random Access Memory. Les applications Twido sont déchargées dans une mémoire RAM volatile interne afin d'être exécutées.
<b>Redémarrage à chaud</b>	Après une coupure secteur, mise sous tension de l'automate sans modification de l'application. L'automate repasse à l'état dans lequel il était avant la coupure secteur et termine la scrutation qui était en cours. Toutes les données de l'application sont préservées. Cette fonction n'est disponible que sur les automates modulaires.
<b>Références croisées</b>	Génération d'une liste d'opérandes, symboles, numéros de ligne/réseau et opérateurs utilisée dans une application pour simplifier la création et la gestion des applications.
<b>Registres</b>	Registres spéciaux internes à l'automate dédiés aux blocs fonction LIFO/FIFO.
<b>Repères</b>	Registres internes de l'automate permettant de stocker les valeurs des variables, constantes, E/S, etc. du programme. Le symbole de pourcentage (%) utilisé en préfixe permet d'identifier les repères. Par exemple, %I0.1 indique un repère dans la RAM de l'automate contenant la valeur de la voie d'entrée 1.
<b>Réseau</b>	Périphériques interconnectés partageant un chemin de données et un protocole de communication communs.

---

<b>Réseau</b>	Un réseau est situé entre deux barres de potentiel d'une grille et se compose d'un groupe d'éléments graphiques reliés entre eux par des liaisons horizontales et verticales. Un réseau peut être constitué au maximum de sept lignes et onze colonnes.
<b>Réseau schéma à contacts/liste d'instructions</b>	Affiche les parties d'un programme en liste d'instructions qui ne sont pas réversibles en langage schéma à contacts.
<b>Routeur</b>	Périphérique connectant au moins deux parties d'un réseau et permettant aux données de circuler entre ces deux parties. Un routeur examine chaque paquet reçu et décide s'il doit le bloquer pour l'isoler du reste du réseau ou s'il doit le transmettre. Le routeur tente d'envoyer le paquet sur le réseau en empruntant le chemin le plus efficace.
<b>RTC</b>	De l'anglais Real-Time Clock. Voir "Horodateur".
<b>RTU</b>	De l'anglais Remote Terminal Unit. Protocole utilisant huit bits permettant la communication entre un automate et un PC.

---

**S**

<b>Sauvegarder</b>	Commande permettant de copier l'application contenue dans la RAM de l'automate à la fois dans la mémoire EEPROM interne de l'automate et dans la cartouche de sauvegarde de mémoire en option (si elle est installée).
<b>Scrutation</b>	Un automate scrute un programme et effectue principalement trois fonctions de base : Il lit d'abord les entrées et place les valeurs correspondantes dans la mémoire. Il exécute ensuite le programme d'application, instruction par instruction, puis il stocke les résultats dans la mémoire. Il utilise enfin les résultats pour mettre à jour les sorties.
<b>Serveur</b>	Processus informatique fournissant des services à des clients. Ce terme peut également désigner le processus informatique hébergeant le service.
<b>Sortie réflexe</b>	En mode comptage, la valeur courante du compteur rapide (%VFC.V) est mesurée en fonction des seuils configurés afin de déterminer l'état des sorties dédiées.
<b>Sorties seuil</b>	Bobines contrôlées directement par le compteur très rapide (%VFC) en fonction des paramètres choisis lors de la configuration.

<b>Sous-réseau</b>	Réseau physique ou logique au sein d'un réseau IP, qui partage une adresse réseau avec d'autres parties du réseau.
<b>Symbole</b>	Chaîne de 32 caractères alphanumériques maximum, dont le premier caractère est alphabétique. Les symboles permettent de personnaliser les objets de l'automate afin de faciliter la maintenance de l'application.
<b>Symbole non résolu</b>	Symbole sans repère de variable.

---

**T**

<b>Table d'animation</b>	Table créée dans un éditeur de langage ou dans un écran d'exploitation. Lorsqu'un PC est connecté à l'automate, la table d'animation permet de visualiser les variables de l'automate et de forcer leurs valeurs lors d'un débogage. Elle peut être enregistrée dans un fichier distinct portant l'extension .tat.
<b>Table de symboles</b>	Table des symboles utilisés dans une application. La table est affichée dans l'éditeur de symboles.
<b>TCP</b>	Protocole de contrôle de transmission (de l'anglais Transmission Control Protocol).
<b>TCP/IP</b>	Suite de protocoles composée du protocole de contrôle de transmission et du protocole Internet. Suite de protocoles de communication sur laquelle repose Internet.
<b>Temporisateur</b>	Bloc fonction utilisé pour sélectionner une durée pour le contrôle d'un événement.
<b>Trame</b>	Groupe de bits constituant un bloc TOR d'informations. Les trames contiennent des informations ou des données de contrôle de réseau. La taille et la composition d'une trame sont définies par la technique de réseau utilisée.
<b>Twido</b>	Gamme d'automates Schneider Electric comprenant deux types d'automate (compacts et modulaires), des modules d'expansion permettant d'ajouter des points d'E/S et des options telles que l'horodateur, les communications, l'afficheur et les cartouches de sauvegarde de mémoire.
<b>TwidoSoft</b>	Logiciel de développement graphique 32 bits fonctionnant sous Windows permettant de configurer et de programmer des automates Twido.
<b>Types de trame</b>	Ethernet II et IEEE 802.3 sont deux types de trame classiques.

---

**V**

<b>Validation auto par ligne</b>	Lors de l'insertion ou de la modification d'instructions en langage liste d'instructions, ce paramètre facultatif permet de valider les lignes de programme à mesure qu'elles sont saisies (recherche des erreurs et des symboles non résolus). Tous les éléments doivent être corrigés pour que le programmeur puisse quitter la ligne. Sélectionné à partir de la boîte de dialogue Préférences.
<b>Variable</b>	Unité de mémoire pouvant être adressée et modifiée par un programme.
<b>Variable de données</b>	Voir "Variable".
<b>Visualiseur de références croisées</b>	Fenêtre spécialisée de l'application TwidoSoft permettant de visualiser les références croisées.
<b>Visualiseur des erreurs du programme</b>	Fenêtre spécialisée de TwidoSoft permettant d'afficher les avertissements et erreurs du programme.

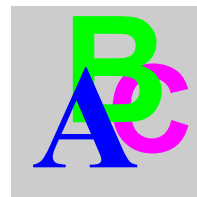
---



---

## Index

---



### Symbols

-	513	%S22	76, 550
%Ci	345	%S23	76, 550
%DR	408	%S24	550
%FC	414	%S31	550
%INW	45	%S38	550
%MSG	432	%S39	550
%PLS	404	%S4	547
%PWM	400	%S5	547
%QNW	45	%S50	551
%S	546	%S51	551
%S0	547	%S52	551
%S1	547	%S59	551
%S10	548	%S6	547
%S100	552	%S66	551
%S101	552	%S69	552
%S103	553	%S7	547
%S104	553	%S75	552
%S11	548	%S8	547
%S110	553	%S9	548
%S111	553	%S95	552
%S112	553	%S96	552
%S113	553	%S97	552
%S118	553	%SBR	351
%S119	554	%SCi	354
%S12	548	%SW	555
%S13	548	%SW0	556
%S17	548	%SW101	565
%S18	549	%SW102	565
%S19	549	%SW103	565
%S20	549	%SW104	565
%S21	76, 549	%SW105	565
		%SW106	565

%SW11, 557  
%SW111, 566  
%SW112, 566  
%SW113, 566  
%SW114, 566  
%SW118, 566  
%SW120, 567  
%SW17, 558  
%SW18, 558  
%SW19, 558  
%SW30, 558  
%SW31, 558  
%SW32, 558  
%SW48, 558  
%SW49, 559  
%SW50, 559  
%SW51, 559  
%SW52, 559  
%SW53, 559  
%SW54, 559  
%SW55, 559  
%SW56, 559  
%SW57, 559  
%SW58, 559  
%SW59, 560  
%SW6, 556  
%SW60, 560  
%SW63, 560  
%SW64, 560  
%SW65, 561  
%SW67, 562  
%SW7, 557  
%SW73, 562  
%SW74, 562  
%SW76, 562  
%SW77, 562  
%SW78, 562  
%SW79, 562  
%SW80, 563  
%SW81..%SW87, 563  
%SW96, 564  
%SW97, 564  
%TM, 342  
%VFC, 418  
\*, 513  
+, 513

/, 513

## A

ABS, 513  
Accès à la configuration  
    PID, 458  
Accès à la mise au point  
    PID, 477  
Accumulateur, 290  
Accumulateur booléen, 290  
ACOS, 518  
Action dérivée, 506  
Action intégrale, 505  
Action proportionnelle, 504  
Adressage de modules d'E/S analogiques, 201  
Adresse de passerelle, 168  
Adresse IP, 167  
    adresse IP par défaut, 169  
Adresse MAC, 169  
Afficheur  
    correction de l'horodateur, 260  
    horloge calendaire, 259  
    ID et états de l'automate, 249  
    paramètres de port série, 258  
    présentation, 246  
    variables et objets système, 251  
Ajouter, 367  
AND, instructions, 324  
ASCII  
    communication, 95  
    communications, 127  
    configuration du port, 130  
    configuration logicielle, 129  
    configuration matérielle, 127  
ASIN, 518  
ATAN, 518

## B

Bit Run/Stop, 78  
Bits mémoire, 27  
Bits système, 546  
BLK, 282



Bloc comparaison  
     élément graphique, 273  
 Bloc fonction %MSG3  
     instruction, 190  
 Bloc fonction compteur rapide (%VFC), 418  
 Bloc fonction compteur rapide (FC), 414  
 Bloc fonction d'échange, 432  
 Bloc fonction programmeur cyclique, 408  
 Blocs  
     dans des schémas à contacts, 268  
 Blocs comparaisons, 269  
 Blocs fonction  
     compteurs, 345  
     élément graphique, 273  
     Fonction pas à pas (%SCi), 354  
     programmeur cyclique, 408  
     programmation de blocs fonction  
         standards, 335  
         PWM, 400  
     registre bits à décalage (%SBR), 351  
     registres, 393  
     temporisateurs, 337, 342  
 Blocs fonction avancés  
     objets mots et objets bits, 388  
 Blocs fonctions  
     blocs horodateurs, 438  
     dans une grille de programmation, 269  
     présentation des blocs fonctions  
         standards, 333  
     programmeur cyclique, 412  
 Blocs fonctions avancés  
     principes de programmation, 390  
 Blocs fonctions standards, 333  
 Blocs opérations, 270  
     élément graphique, 273  
 Bobines, 268  
     éléments graphiques, 272  
 Brochages  
     connecteur femelle du câble de  
         communication, 99  
     connecteur mâle du câble de  
         communication, 99  
 Bus AS-Interface V2  
     adressage automatique d'un esclave,

234  
     adressage des E/S, 237  
     changement d'adresse d'un esclave, 227  
     configuration logicielle, 218  
     description fonctionnelle générale, 211  
     diagnostic des esclaves, 226  
     échanges explicites, 239  
     échanges implicites, 238  
     écran de configuration, 216  
     écran de mise au point, 224  
     esclave défectueux, 236  
     insertion esclave, 235  
     mise au point du bus, 229  
     mode de fonctionnement, 244  
     présentation, 210  
     principe de mise en œuvre logicielle, 214  
     prise en compte nouvelle configuration,  
         232  
     programmation et diagnostic du bus AS-  
         Interface, 239  
     transfert de l'image d'un esclave, 230

## C

Calcul, 367  
 Caractéristiques du PID, 454  
 Chaînes de bits, 48  
 Chien de garde logiciel, 73  
 Commentaires de lignes Liste, 284  
 Communication par modem, 97  
 Communications  
     ASCII, 127  
     liaison distante, 114  
     Modbus, 138  
 Compteurs, 345  
     Programmation et configuration, 349  
 Configuration  
     PID, 458  
     port pour ASCII, 130  
     port pour Modbus, 141  
     table d'émission/réception pour ASCII,  
         130  
 Configuration TCP/IP, 173  
 Conseils de programmation, 275  
 Contacts, 268  
     élément graphique, 271

Contrôle d'inactivité, 179  
Correction du RTC, 437  
COS, 518  
Coupure secteur, 77  
Cycle de tâche, 73

## D

Débordement, 368  
    index, 53  
Débordement d'index, 53  
Décrément, 367  
DEG\_TO\_RAD, 521  
Démarrage à froid, 82  
Détection de fronts  
    descendants, 317  
    montants, 316  
DINT\_TO\_REAL, 523  
Diviser, 367  
Documentation de votre programme, 284  
Durée de scrutation, 73

## E

E/S  
    repérage, 43  
Éléments de liaison  
    éléments graphiques, 271  
Éléments graphiques  
    schémas à contacts, 271  
Émission de messages, 431  
END\_BLK, 282  
En-tête réseau, 267  
    commentaires, 285  
EQUAL\_ARR, 529  
Erreur, 368  
Ethernet  
    configuration TCP/IP, 173  
    connexion réseau, 166  
    gestion des connexions, 185  
EXCH, 431  
EXCH3, 190  
    code d'erreur, 193  
Exemple  
    Compteur/Décompteur, 350  
EXP, 513

EXPT, 513

## F

Facteur de correction de l'horodateur, 260  
FIFO  
    fonctionnement, 396  
    introduction, 393  
File d'attente, 393  
FIND\_, 531  
Fonction pas à pas, 354  
Fonctions horloges  
    horodatage, 441  
    présentation, 437  
    réglage de la date et de l'heure, 443  
Fonctions horodateurs  
    blocs horodateurs, 438

## G

Génération d'impulsions, 404  
Gestion des connexions, 185  
Grafcet  
    actions associées, 308  
    exemples, 302  
    instructions, 300  
    pré-traitement, 305  
    traitement séquentiel, 306  
Grafcet, méthodes, 75  
Grille de programmation, 266

## I

ID unité, 182  
Incrément, 367  
Initialisation des objets, 84  
Instruction EXCH, 431

**Instructions**

- AND, 324
- arithmétiques, 367
- chargement, 320
- de comparaison, 365
- de conversion, 374
- END, 378
- JMP, 381
- logiques, 370
- NOP, 380
- NOT, 330
- RET, 382
- SR, 382
- XOR, 328

**Instructions arithmétiques, 367****Instructions booléennes, 316**

- affectation, 322
- explication du format utilisé dans ce manuel, 318
- OR, 326

**Instructions d'affectation**

- numériques, 360

**Instructions de affectation, 322****Instructions de comparaison, 365****Instructions de conversion, 374****Instructions de conversion entre mots**

- simples et doubles, 376

**Instructions de décalage, 372****Instructions de pile, 297****Instructions de saut, 381****Instructions de sous-programme, 382****Instructions en langage liste d'instructions, 291****Instructions END, 378****Instructions logiques, 370****Instructions numériques**

- affectation, 360
- de décalage, 372

**INT\_TO\_REAL, 523****IP repérée, 177****J**

- JMP, 381

**L**

- LAN ACT, 188

- LAN ST, 189

- Langage liste d'instructions

- vue d'ensemble, 288

- Langages de programmation

- présentation, 21

- LD, 320

- LDF, 317, 320

- LDN, 320

- LDR, 316, 320

- Liaison ASCII

- exemple, 135

- Liaison distante

- accès aux données E/S distantes, 120

- communication, 95

- communications, 114

- configuration de l'automate distant, 118

- configuration de l'automate maître, 117

- configuration logicielle, 117

- configuration matérielle, 115

- exemple, 123

- synchronisation de scrutation de

- l'automate distant, 119

- Liaison Modbus

- exemple 1, 147

- exemple 2, 150

- LIFO

- fonctionnement, 395

- introduction, 393

- LKUP, 538

- LN, 513

- LOG, 513

**M**

- Masque de sous-réseau, 167

- MAX\_ARR, 533

- MEAN, 543

- Mémoire

- cartouche de 32 K, 61

- cartouche de 64 K, 64

- sans cartouche, 59

- structure, 56

- MIN\_ARR, 533

**Mise au point**

PID, 477

**Modbus**

communication, 96

communications, 138

configuration du port, 141

configuration logicielle, 140

configuration matérielle, 138

esclave, 96

maître, 96

messagerie Modbus TCP, 190

requêtes standard, 153

TCP Client/Serveur, 159

**Modbus TCP/IP**

périphériques distants, 181

**Modes de fonctionnement, 75****Modulation de la largeur d'impulsion, 400****Module analogique**

exemple, 206

fonctionnement, 200

**Modules analogiques**

adressage, 201

configuration d'E/S, 203

mots mémoire, 30

Mots système, 555

MPP, 297

MPS, 297

MRD, 297

Multiplier, 367

**N**

NOP, 380

NOP, instruction, 380

NOT, instruction, 330

**O****Objets**

blocs fonction, 46

double mot, 34

flottant, 34

mots, 30

objets bits, 27

structurés, 48

**Objets bits, 388**

adressage, 39

présentation, 27

**Objets flottants**

adressage, 41

vue d'ensemble, 34

**Objets mots, 388**

adressage, 40

vue d'ensemble, 30

**Objets mots doubles, 47**

adressage, 42

vue d'ensemble, 34

**OCCUR\_ARR, 534****Onglet Animation**

PID, 479

**Onglet Auto tuning**

PID, 469

**Onglet Entrée**

PID, 463

**Onglet Général**

PID, 460

**Onglet PID**

PID, 466

**Onglet Sortie**

PID, 474

**Onglet Trace**

PID, 482

**OPEN, 274****Opérandes, 290****OR exclusif, instructions, 328****OR, instruction, 326****OUT\_BLK, 282****P****Paramètres, 338****Paramètres de contrôle**

ASCII, 130

**Parenthèses**

imbrication, 295

modificateurs, 295

utilisation dans des programmes, 294

**PID**

- configuration, 458
- mise au point, 477
- onglet Animation, 479
- onglet Auto tuning, 469
- onglet Entrée, 463
- onglet Général, 460
- onglet PID, 466
- onglet Sortie, 474
- onglet Trace, 482
- présentation générale, 450

Pile, 393

Point de réglage, 196

Présentation des communications, 95

Présentation générale

- PID, 450

Programamtion non réversible, 390

Programmateurs cycliques

- programmation et configuration, 412

Programmation

- documentation de votre programme, 284

Programmation réversible, 390

Programme par schémas à contacts

- conversion en liste d'instructions, 280

Programming Principles, 390

Protocole

- modbus TCP/IP, 96

Protocoles, 95

**R**

Raccordement du câble de communication, 97

Racine carrée, 367

RAD\_TO\_DEG, 521

REAL\_TO\_DINT, 523

REAL\_TO\_INT, 523

Réception de messages, 431

Registers

- FIFO, 396

Registre bits à décalage, 351

Registres

- LIFO, 395

- programmation et configuration, 397

Réglage en boucle fermée, 501

Réglage en boucle ouverte, 502

Repérage

- objets indexés, 52

Repérage des E/S, 43

Repérage direct, 52

Reprise à chaud, 80

Reprise secteur, 77

Réseau

- adressage, 45

Réseau schéma à contacts / liste

d'instructions, 283

Réseaux

- inconditionnels, 283

Réseaux inconditionnels, 283

Réseaux schéma à contacts, 265

Reste, 367

RET, 382

Réversibilité

- introduction, 280

- recommandations, 282

ROL\_ARR, 535

ROR\_ARR, 535

**S**

Sauvegarde et restauration

- cartouche de mémoire étendue de 64 K, 64

- cartouche de sauvegarde de 32 K, 61
- sans cartouche, 59

- structure de la mémoire, 56

Schémas à contacts

- blocs, 268

- éléments graphiques, 271

- introduction, 264

- OPEN et SHORT, 274

- principes de programmation, 266

Scrutation

- cyclique, 68

- périodique, 70

SHORT, 274

SIN, 518

SORT\_ARR, 537

Soustraire, 367

SQRT, 513

SR, 382

SUM\_ARR, 527

Symbolisation, 54

Zone de test, 266

## T

Table de contrôle

Modbus, 142

Tables d'objets, 48

Tâches événementielles

gestion des événements, 89

les différentes sources d'événement, 87

présentation, 86

TAN, 518

TCP Client/Serveur, 159

TCP/IP

protocole, 96

Temporisateurs, 338

base temps de 1 ms, 343

introduction, 337

programmation et configuration, 342

TOF, type, 339

TON, type, 340

TP, type, 341

TOF, temporisateur, 339

TON, temporisateur, 340

TP, type de temporisateur, 341

Traitement numérique

Présentation, 359

TRUNC, 513

TwidoSoft

introduction, 20

## V

Valeur absolue, 367

Validation d'objets, 26

Vérification de la durée de scrutation, 73

Voie analogique, 198

## X

XOR, 328

## Z

Zone d'action, 266